



**UNICAMP**

UNIVERSIDADE ESTADUAL DE  
CAMPINAS

Instituto de Matemática, Estatística e  
Computação Científica

IVAN XAVIER MOURA DO NASCIMENTO

**Sobre o Problema da Decomposição Matricial  
em Componentes Esparsa e de Posto Baixo  
com Exemplos em Videovigilância**

Campinas

2021

Ivan Xavier Moura do Nascimento

# Sobre o Problema da Decomposição Matricial em Componentes Esparsa e de Posto Baixo com Exemplos em Videovigilância

Tese apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Doutor em Matemática Aplicada.

Orientadora: Sandra Augusta Santos

Coorientador: Paulo José da Silva e Silva

Este trabalho corresponde à versão final da Tese defendida pelo aluno Ivan Xavier Moura do Nascimento e orientada pela Profa. Dra. Sandra Augusta Santos.

Campinas

2021

Ficha catalográfica  
Universidade Estadual de Campinas  
Biblioteca do Instituto de Matemática, Estatística e Computação Científica  
Sylvania Renata de Jesus Ribeiro - CRB 8/6592

N17s Nascimento, Ivan Xavier Moura do, 1989-  
Sobre o problema da decomposição matricial em componentes esparsa e de posto baixo com exemplos em videovigilância / Ivan Xavier Moura do Nascimento. – Campinas, SP : [s.n.], 2021.

Orientador: Sandra Augusta Santos.  
Coorientador: Paulo José da Silva e Silva.  
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1. Otimização convexa. 2. Suavização (Análise numérica). 3. Continuação homotópica. 4. Método de gradiente proximal. I. Santos, Sandra Augusta, 1964-. II. Silva, Paulo José da Silva e, 1972-. III. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. IV. Título.

Informações para Biblioteca Digital

**Título em outro idioma:** On the sparse and low rank matrix decomposition problem with examples in video surveillance

**Palavras-chave em inglês:**

Convex optimization

Smoothing (Numerical analysis)

Homotopy continuation

Proximal gradient method

**Área de concentração:** Matemática Aplicada

**Titulação:** Doutor em Matemática Aplicada

**Banca examinadora:**

Sandra Augusta Santos [Orientador]

Renato da Rocha Lopes

José Mario Martínez Pérez

Elias Salomão Helou Neto

Roberto Hirata Junior

**Data de defesa:** 29-11-2021

**Programa de Pós-Graduação:** Matemática Aplicada

**Identificação e informações acadêmicas do(a) aluno(a)**

- ORCID do autor: <https://orcid.org/0000-0001-8464-6068>

- Currículo Lattes do autor: <http://lattes.cnpq.br/9651090989298780>

**Tese de Doutorado defendida em 29 de novembro de 2021 e aprovada  
pela banca examinadora composta pelos Profs. Drs.**

**Prof(a). Dr(a). SANDRA AUGUSTA SANTOS**

**Prof(a). Dr(a). RENATO DA ROCHA LOPES**

**Prof(a). Dr(a). JOSÉ MARIO MARTÍNEZ PÉREZ**

**Prof(a). Dr(a). ELIAS SALOMÃO HELOU NETO**

**Prof(a). Dr(a). ROBERTO HIRATA JUNIOR**

A Ata da Defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria de Pós-Graduação do Instituto de Matemática, Estatística e Computação Científica.

*À Dona Sonia,  
minha amada mãe,  
e à Vó Matilde,  
uma das mais de  
619 mil vítimas fatais  
da Covid-19  
no Brasil até o momento.*

# AGRADECIMENTOS

*“O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001”. Também viabilizou apoio financeiro o Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), processo número 140894/2016-0, Chamada/Edital: GM/GD – Cotas do Programa de Pós-Graduação.*

À Sandra, por uma década de colaboração em que transbordou excelência acadêmica e humanidade de maneira ímpar e inspiradora. Ao Paulo, pelas valiosas contribuições mesmo antes da coorientação, ainda nos seminários do grupo, pelo apoio perene e também pelo crescente cuidado e zelo que demonstrou para comigo. A ambos pelo empenho em disponibilizar a estrutura presencial e remota do Laboratório de Otimização Contínua (LOC), sem a qual tudo teria sido muito mais difícil nestes tempos pandêmicos; e, principalmente, por acreditarem em mim mesmo quando eu duvidava de que seria capaz de concluir o trabalho.

Ao Professor Jon Lee e à Professora Marcia Fampa por serem corresponsáveis pelo meu primeiro contato com o tema do trabalho e pelos breves e divertidos momentos compartilhados presencialmente.

Aos Professores Elias S. Helou Neto, Mario Martínez, Renato R. Lopes e Roberto Hirata Jr. por comporem a banca da defesa e por todas as sugestões e apontamentos que tanto contribuíram para a versão final desta tese. Aos Professores Ernesto Birgin, Lucio T. dos Santos e Luis Felipe Bueno por aceitarem a tarefa inglória porém indispensável da suplência. Em especial, ao Lucio e ao Mario pelo carinho e pelos ensinamentos ao longo dos meus quinze anos de IMECC, entre idas e vindas.

Ao corpo docente e demais funcionários do instituto, além dos meus colegas discentes, dos quais destaco os do Laboratório de Pesquisa Operacional e Otimização (LPOO), sob responsabilidade do Professor Aurelio, por me acolherem durante um bom tempo, e também os meus colegas do LOC. Aos Professores Gilli e Cheti pela confiança depositada em mim por meio de suas cartas de recomendação para o ingresso no programa de doutorado. À Dona Zefa pelos incontáveis cafés e abraços. Ao Fábio R. Silva pelo modelo de dissertações e teses em  $\text{\LaTeX}/\text{abnTeX2}$ , e às bibliotecárias Sylvania Ribeiro e Ana Regina Machado pela ficha catalográfica.

À Unicamp por viabilizar minha vida acadêmica nos meus primeiros anos de graduação com seus imprescindíveis auxílios de permanência estudantil. À nossa comunidade universitária, sobretudo aos seus trabalhadores – docentes, servidores

efetivos e funcionários terceirizados – que dia-a-dia se dedicam e constroem essa instituição, tornando possível que muitas histórias sejam escritas e vivenciadas dentro e fora dos *campi*.

À Universidad de Santiago de Chile, às ETECs Escolástica Rosa e Aristoteles Ferreira, às escolas Brás Cubas e Emílio Justo, além do Lar Veneranda e da Gota de Leite, porque há um pouco de cada uma dessas instituições em mim e, conseqüentemente, neste trabalho.

Aos meus professores do Fundamental e do Médio pela coragem e perseverança como docentes do ensino público. Em particular, às professoras Neuza, Maria Laura, Sumara, Maria Angélica, Adenilda, Mônica, Rosângela, Deise, Rosa, Denise e Fernanda.

À minha mãe, Dona Sonia, pelo apoio incondicional de sempre, inclusive quando decidi sair de casa para estudar a 200 quilômetros de distância, ainda antes da maioridade e com pouquíssimos recursos, e apesar de eu nunca mais ter voltado em definitivo. À Karen, minha eterna irmãzinha, quem vi brotar e desabrochar diariamente nos primeiros sete anos de vida; à irmã caçula, Angélica; e aos irmãos mais velhos, Cris, Kelly e Sidney, por entenderem minhas ausências e por preservarem com afeto nossas memórias de infância e adolescência.

Aos meus queridos amigos e amores, em particular aos da Moradia Estudantil, do Barraco e do El Punto, além de seus inúmeros agregados, por tanta vida compartilhada nessa linda jornada de estudante universitário que supostamente aqui se encerra. À Déborah e ao Nelson, por serem há tanto tempo minha segunda família; ao Felipe e ao André, pelo companheirismo dos últimos anos, e ao Augusto, por toda a cumplicidade e dedicação, principalmente em tempos tão estranhos e difíceis. À Carol, ao Cot e à Thirza, pela façanha mútua de nutrir relações com quem tão pouco se vê pessoalmente.

Por fim, aos cientistas e governantes que batalharam pelas vacinas contra o coronavírus, assim como aos profissionais de saúde e trabalhadores essenciais, cujas atuações impactaram profunda e extraordinariamente todos os aspectos das nossas vidas nos últimos dois anos, em que pese o esforço cotidiano do governo federal em aniquilar o povo brasileiro, sob o comando do ser abjeto e ignóbil que desgraçadamente ora preside o país – mas, como tudo indica, não por muito mais tempo.

# RESUMO

Neste trabalho, tratamos do problema discreto fundamental de decomposição de uma dada matriz retangular como a soma de uma parcela esparsa com outra de posto baixo, aplicado a situações prototípicas de videovigilância, sem informações prévias sobre padrões de esparsidade e posto subjacentes.

Como é comum na literatura, esparsidade e posto baixo são induzidos nas componentes por meio das funções de aproximação que constituem os respectivos envelopes convexos: a norma 1 do vetor de entradas matriciais e a norma nuclear matricial. A combinação convexa de tais normas constitui então uma função objetivo a ser minimizada sob restrições de igualdade.

Nossa proposta consiste na suavização do termo da norma 1 via função de Huber e na incorporação das restrições à nova função objetivo, a qual é otimizada com um método do tipo gradiente proximal com tamanho do passo *à la* Barzilai-Borwein associado a um esquema de continuação homotópica que atua no ajuste do parâmetro de suavização.

Conduzimos experimentos numéricos com instâncias de filmagens reais, de porte da ordem de dezenas de milhões de variáveis, em concorrência com o Método dos Multiplicadores com Direções Alternadas (ADMM), estado-da-arte para a classe de problemas.

Os resultados mostraram que, no que diz respeito à reconstrução das imagens, nossa abordagem atinge precisão suficiente com menos esforço computacional, em termos do número de decomposições em valores singulares. Além disso, desenvolvemos uma heurística para a escolha do ponto inicial que acelerou substancialmente a convergência do ADMM nas situações práticas analisadas.

**Palavras-chave:** decomposição matricial, esparsidade e posto baixo, gradiente proximal, análise de componentes principais, videovigilância

# ABSTRACT

In this work, we tackle the fundamental discrete problem of decomposing a given rectangular matrix as the sum of a sparse and a low-rank components, applied to prototypical video surveillance instances, without prior information on the underlying sparsity and rank patterns.

As it is common in the literature, sparsity and low rank are induced by the respective convex hull approximation functions: the vector 1-norm and the nuclear norm. The convex combination of such norms then constitutes an objective function to be minimized under equality constraints.

Our proposal consists in smoothing the 1-norm using the Huber function and in incorporating the constraints into the new objective function, which is optimized with a proximal gradient-type method with the Barzilai-Borwein stepsize, combined with a homotopy continuation scheme that acts on the adjustment of the smoothing parameter.

We conducted numerical experiments with instances of real footage, with tens of millions of variables, and compared the results with the Alternating Direction Method of Multipliers (ADMM), one of the state-of-the-art algorithms for this class of problems.

The results showed that, with regard to visual image reconstruction, our approach achieves sufficient precision with less computational effort, in terms of the number of Singular Value Decompositions. In addition, we developed a warm-start heuristic that substantially accelerated ADMM convergence in the practical instances.

**Keywords:** matrix decomposition, sparsity and low rankness, proximal gradient, principal component analysis, video surveillance

# LISTA DE ILUSTRAÇÕES

Figura 1 – Condições Suficientes de Unicidade de Solução . . . . .	42
Figura 2 – Aproximações Suaves para o Valor Absoluto . . . . .	51
Figura 3 – Aleatórias: Distribuição das Entradas Matriciais . . . . .	82
Figura 4 – Aleatórias: Distribuição dos Valores Singulares . . . . .	84
Figura 5 – Aleatórias: Erros Relativos das Aproximações $A_k$ (n=1000) . . . . .	89
Figura 6 – Aleatórias: Erros Relativos, ADMM <i>vs</i> SPGM (1a Bateria) . . . . .	96
Figura 7 – Aleatórias: SPGM, Instâncias Seleccionadas (1a Bateria) . . . . .	98
Figura 8 – Aleatórias: SPGM, Comparação de Configurações . . . . .	101
Figura 9 – Aleatórias: <i>Data Profiles</i> das Configurações . . . . .	105
Figura 10 – Conversão de Frames em Matrizes SLRMD . . . . .	109
Figura 11 – Vídeos CAVIAR: Frames Originais e <i>Ground-Truths</i> . . . . .	125
Figura 12 – Vídeos SBI: Frames Originais e <i>Ground-Truths</i> . . . . .	126
Figura 13 – Vídeos CAVIAR: Soluções de Referência . . . . .	128
Figura 14 – Aleatórias: Grupo SBI, Solução de Referência . . . . .	129
Figura 15 – Vídeos: Distribuição das Entradas . . . . .	133
Figura 16 – Vídeos: Distribuição dos Valores Singulares . . . . .	136
Figura 17 – Vídeos: Distribuição dos Valores Singulares . . . . .	137
Figura 18 – Vídeos: Grupo CAVIAR, SPGM <i>vs</i> ADMM . . . . .	141
Figura 19 – Vídeos: Grupo SBI, SPGM <i>vs</i> ADMM . . . . .	142
Figura 20 – Vídeos: <i>Data Profiles</i> , ADMM <i>vs</i> SPGM . . . . .	147
Figura 21 – Vídeos: Grupo CAVIAR, Medidas no Espaço de Exibição . . . . .	150
Figura 22 – Vídeos: Grupo SBI, Medidas no Espaço de Exibição . . . . .	151
Figura 23 – Aleatórias: Erro Relativo com e sem <i>Warm-start</i> . . . . .	158
Figura 24 – Aleatórias: <i>Data Profiles</i> com e sem <i>warm-start</i> . . . . .	160
Figura 25 – Vídeos: Grupo CAVIAR, ADMM <i>vs</i> SPGM, com e sem <i>warm-start</i> . . . . .	162
Figura 26 – Vídeos: Grupo SBI, ADMM <i>vs</i> SPGM, com e sem <i>warm-start</i> . . . . .	163
Figura 27 – Vídeos: <i>Data Profiles</i> , ADMM <i>vs</i> . SPGM . . . . .	165
Figura 28 – Vídeos: CAVIAR, Comparação de Frames Seleccionados . . . . .	168
Figura 29 – Vídeos: SBI, Comparação de Frames Seleccionados . . . . .	169
Figura 30 – Vídeos: Grupo CAVIAR, Espaço de Exibição com <i>Warm-Start</i> . . . . .	219
Figura 31 – Vídeos: Grupo SBI, Espaço de Exibição com <i>Warm-Start</i> . . . . .	220

# LISTA DE TABELAS

Tabela 1 – Instâncias de Teste Aleatórias: correspondência rótulo-instância. . . . .	76
Tabela 2 – Parâmetros de Validação do ADMM . . . . .	78
Tabela 3 – Instâncias aleatórias impulsivas baseadas em Candès <i>et al.</i> ([21]). . . . .	79
Tabela 4 – Instâncias aleatórias impulsivas baseadas em Yuan e Yang ([103]). . . . .	79
Tabela 5 – Instâncias aleatórias Gaussianas baseadas em Yuan e Yang ([103]). . . . .	80
Tabela 6 – Aleatórias: Erros Relativos das Soluções Triviais . . . . .	91
Tabela 7 – Parâmetros do SPGM: Bateria 1 (Ausência de Homotopia). . . . .	95
Tabela 8 – Vídeos: Descrição das Cenas do Grupo Caviar . . . . .	121
Tabela 9 – Vídeos: Grupo CAVIAR, Recortes das Cenas . . . . .	122
Tabela 10 – Vídeos: Grupo SBI, Descrição das Cenas . . . . .	123
Tabela 11 – Vídeos: Grupo SBI, Recortes das Cenas . . . . .	124
Tabela 12 – ADMM: Parâmetros para Solução de Referência . . . . .	125
Tabela 13 – Vídeos: Grupo CAVIAR, Resultados da Solução de Referência . . . . .	130
Tabela 14 – Vídeos: Grupo SBI, Resultados das Solução de Referência . . . . .	130
Tabela 15 – Vídeos: ADMM, Parâmetros de Resolução . . . . .	140
Tabela 16 – Vídeos: SPGM, Parâmetros de Resolução . . . . .	140
Tabela 17 – Vídeos: Tabela de Problemas Resolvidos . . . . .	146
Tabela 18 – SPGM: Parâmetros com e sem <i>Warm-Start</i> . . . . .	157
Tabela 19 – Parâmetros do estudo de desempenho do ADMM com e sem <i>warm-start</i>	157
Tabela 20 – Vídeos: Tabela de Problemas Resolvidos com <i>Warm-Start</i> . . . . .	165
Tabela 21 – Aleatórias: CLMW, Esparsidades . . . . .	190
Tabela 22 – Aleatórias: CLMW, Postos . . . . .	191
Tabela 23 – Aleatórias: YImp, Esparsidades . . . . .	192
Tabela 24 – Aleatórias: YImp, Postos . . . . .	193
Tabela 25 – Aleatórias: YGaus, Esparsidades . . . . .	194
Tabela 26 – Aleatórias: YGaus, Postos . . . . .	195
Tabela 27 – Vídeos: Esparsidade de Entrada e da Solução . . . . .	197
Tabela 28 – Vídeos: CAVIAR, Posto de Entrada e da Solução . . . . .	198
Tabela 29 – Vídeos: Grupo SBI, Esparsidade de Entrada e da Solução . . . . .	199
Tabela 30 – Vídeos: Grupo SBI, Posto de Entrada e da Solução . . . . .	200
Tabela 31 – Vídeos: Grupo CAVIAR, Espaço de Exibição . . . . .	210
Tabela 32 – Vídeos: Grupo SBI, Espaço de Exibição . . . . .	211
Tabela 33 – Vídeos: Grupo CAVIAR, Espaço de Exibição com <i>Warm-Start</i> . . . . .	213
Tabela 34 – Vídeos: Grupo SBI, Espaço de Exibição com <i>Warm-Start</i> . . . . .	214
Tabela 35 – Vídeos: Grupo CAVIAR, Dados Iniciais com e sem <i>Warm-Start</i> . . . . .	216
Tabela 36 – Vídeos: Grupo SBI, Dados Iniciais com e sem <i>Warm-Start</i> . . . . .	217

Tabela 37 – Tabela reproduzida e adaptada de [103, Tabela 1], com tradução livre. “Resultados recuperados: $A^*$ é esparsa impulsiva” . . . . .	222
Tabela 38 – Tabela reproduzida e adaptada de [103, Tabela 2], com tradução livre. “Resultados recuperados: $A^*$ é esparsa Gaussiana” . . . . .	222
Tabela 39 – Tabela reproduzida e adaptada de [21, Tabela I], com tradução livre. “Recuperação Correta para Problemas Aleatórios com Variação de Tamanho” . . . . .	223

# LISTA DE ALGORITMOS

Algoritmo 1 – Método dos Multiplicadores com Direções Alternadas (ADMM)	58
Algoritmo 2 – Método do Gradiente Proximal Espectral (SPGM) . . . . .	64
Algoritmo 3 – SPGM com Continuação Homotópica . . . . .	65
Algoritmo 4 – Heurística para o Ponto Inicial . . . . .	156
Algoritmo 5 – Construção de Instância Aleatória de Candès <i>et al.</i> no MATLAB	187
Algoritmo 6 – Construção de Instância Impulsiva de Yuan e Yang no MATLAB	187
Algoritmo 7 – Construção de Instância Gaussiana de Yuan e Yang no MATLAB	188

# LISTA DE SÍMBOLOS

$\stackrel{\text{def}}{=}$	Igualdade por definição
$\lesssim$	Aproximação por valores menores (subaproximação)
$\emptyset$	Conjunto vazio
$E^c$	Conjunto complementar ao conjunto $E$
$E^\perp$	Complemento ortogonal ao espaço $E$
$\text{supp}$	Conjunto de suporte de uma função
$\text{sign}$	Função sinal a valores -1, 0 ou 1
$f^*(y)$	Função conjugada convexa de $f(x)$
$\nabla f$	Gradiente da função $f$
$\langle \cdot, \cdot \rangle$	Produto interno
$\text{diag}(A)$	Vetor coluna que coincide com a diagonal da matriz $A$
$\text{Diag}(v)$	Matriz cuja diagonal principal coincide com o vetor $v$
$I_n$	Matriz identidade de dimensão $n \times n$
$\text{tr}(A)$	Traço da matriz $A$
$[A]_+$	Matriz das partes positivas das entradas da matriz $A$
$\  \cdot \ _1$	Norma 1 vetorial
$\  \cdot \ _2$	Norma 2 vetorial ou norma espectral matricial (maior valor singular)
$\  \cdot \ _\infty$	Norma infinito vetorial (maior componente em módulo)
$\  \cdot \ _F$	Norma de Frobenius matricial
$\  \cdot \ _\star$	Norma de nuclear matricial (soma dos valores singulares)
$\  \cdot \ _d$	Norma dual de uma norma $\  \cdot \ $

# SUMÁRIO

<b>I</b>	<b>Introdução</b>	<b>18</b>
<b>1</b>	<b>Introdução</b>	<b>19</b>
1.1	A decomposição matricial e a separação de imagens	19
1.2	RPCA e VCA	23
1.3	Nossa abordagem para o problema	23
1.3.1	Remodelagem e Resolução Numérica	24
1.3.2	Métodos Semelhantes	25
1.3.3	Metodologia de comparação	25
1.3.4	Contribuições deste trabalho	26
1.4	Algumas Notações	26
1.5	Organização do Trabalho	28
<b>II</b>	<b>Teoria e Métodos Computacionais</b>	<b>29</b>
<b>2</b>	<b>Fundamentos Teóricos</b>	<b>30</b>
2.1	Formulações Matemáticas	30
2.2	Convexidade Não Estricta	32
2.3	Dualidade	33
2.4	Condições de Otimalidade	35
2.5	Condições suficientes de otimalidade (Chandrasekaran <i>et al.</i> )	39
2.5.1	Algumas considerações	42
2.6	Caracterização via Programação Semidefinida	44
2.7	Aproximações Suaves para as Normas via Dualidade	47
2.7.1	Aproximação Suave de Primeira Ordem: a função de Huber	48
2.7.1.1	Aproximação de Classe $C^1$	49
2.7.2	Aproximação de Segunda Ordem: a função pseudo-Huber	50
2.7.3	Derivadas das Aproximações Suaves	51
2.7.4	Alguns comentários	52
2.8	Melhores Aproximações para uma Matriz via SVD	52
<b>3</b>	<b>Métodos Computacionais</b>	<b>55</b>
3.1	Método dos Multiplicadores com Direções Alternadas (ADMM)	55
3.2	Método do Gradiente Proximal Espectral (SPGM)	57
3.2.1	O modelo local para a função objetivo	59
3.2.2	Crítério não monótono de decréscimo suficiente	60
3.2.3	Suavização da norma 1 e operador SVT	61
3.3	Método do Gradiente Proximal Espectral: Algoritmos	61

3.3.1	Formulação do Problema . . . . .	61
3.3.2	Algoritmos . . . . .	64
<b>III</b>	<b>Experimentos Computacionais</b>	<b>66</b>
<b>4</b>	<b>Experimentos Numéricos com Dados Artificiais Aleatórios</b> . . . . .	<b>68</b>
4.1	Algumas Definições, Notações e Nomenclaturas Preliminares . . . . .	68
4.2	Construção das Instâncias Aleatórias . . . . .	73
4.3	Validação da implementação do ADMM . . . . .	77
4.4	Variabilidade nas Instâncias de Teste . . . . .	81
4.4.1	Entradas Matriciais . . . . .	82
4.4.2	Valores Singulares Matriciais . . . . .	83
4.4.3	Relação Entradas e Valores Singulares . . . . .	85
4.5	Aproximações de posto fixo como solução das aleatórias . . . . .	87
4.6	Experimentos Preliminares de Calibração com o SPGM . . . . .	92
4.7	Comparação de Desempenho nas Instâncias Aleatórias . . . . .	102
<b>5</b>	<b>Vídeos, Espaço de Exibição e Precisão-Alvo</b> . . . . .	<b>106</b>
5.1	Dos vídeos aos Iterandos . . . . .	106
5.2	Espaços de Otimização e de Exibição . . . . .	110
5.3	Soluções de Referência . . . . .	115
5.4	Medições Auxiliares . . . . .	116
5.5	Precisão-Alvo . . . . .	117
<b>6</b>	<b>Instâncias Práticas de Teste: Imagens de Videovigilância</b> . . . . .	<b>120</b>
6.1	Problemas-testes de Videovigilância . . . . .	120
6.2	Soluções de Referência para os Vídeos . . . . .	124
6.3	Análise Estrutural das Matrizes dos Vídeos . . . . .	131
6.3.1	Entradas Matriciais . . . . .	131
6.3.2	Valores Singulares . . . . .	134
6.3.3	Comparação com Matrizes Aleatórias . . . . .	135
<b>7</b>	<b>Comparação de Desempenho nas Instâncias dos Vídeos</b> . . . . .	<b>139</b>
7.1	Erros Relativos às Soluções de Referência . . . . .	139
7.2	Convergência Segundo a Precisão-alvo . . . . .	145
7.3	Medidas no Espaço de Exibição . . . . .	147
7.4	Considerações Finais . . . . .	152
<b>8</b>	<b>Discussões sobre o Ponto Inicial</b> . . . . .	<b>154</b>
8.1	<i>Warm-start</i> : Uma Heurística de Partida Aprimorada . . . . .	154
8.1.1	Resultados Numéricos da Partida Aprimorada . . . . .	157
8.1.1.1	Instâncias Aleatórias . . . . .	157
8.1.1.2	Instâncias de Videovigilância . . . . .	161
8.2	Pontos Iniciais no Espaço de Exibição . . . . .	166

<b>IV Considerações Finais</b>	<b>171</b>
9 Considerações Finais . . . . .	172
<b>REFERÊNCIAS</b> . . . . .	<b>174</b>
<b>Apêndices</b>	<b>183</b>
APÊNDICE A Algumas contas . . . . .	184
APÊNDICE B Códigos de Geração das Instâncias Aleatórias no MATLAB	187
APÊNDICE C Dados de Esparsidade e Posto das Instâncias Aleatórias . .	189
APÊNDICE D Dados de Esparsidade e Posto das Instâncias dos Vídeos . .	196
APÊNDICE E Imagens e Vídeos Digitais . . . . .	201
APÊNDICE F Resultados Finais: Vídeos sem <i>Warm-Start</i> . . . . .	209
APÊNDICE G Resultados Finais: Vídeos com <i>Warm-Start</i> . . . . .	212
APÊNDICE H Tabelas com Dados Iniciais dos Vídeos . . . . .	215
APÊNDICE I Gráficos ER, PEP, EMPE: Vídeos com <i>Warm-Start</i> . . . . .	218
<b>Anexos</b>	<b>221</b>
ANEXO A Tabelas Publicadas com Resultados do ADMM . . . . .	222

## Parte I

### Introdução

# 1 INTRODUÇÃO

Neste capítulo, apresentamos o contexto do trabalho desenvolvido e a revisão bibliográfica dos principais temas relacionados, assim como introduzimos o problema de pesquisa, delimitamos o seu escopo e antecipamos os principais resultados e contribuições.

## 1.1 A decomposição matricial e a separação de imagens

A adoção de estruturas matriciais na modelagem de problemas de grande porte e alta complexidade está amplamente difundida em diversas áreas de pesquisa e tecnologia. É comum que se necessite destrinchar dados armazenados matricialmente visando extrair informações que ajudem a solucionar o problema de partida. Uma dessas situações, alvo deste trabalho, é aquela em que uma matriz precisa ser decomposta na soma de outras duas de mesma dimensão, de maneira que uma delas seja esparsa e a outra tenha posto baixo.

O estudo de tais problemas tem ampla relevância, na medida em que aplicações em diversas áreas podem ser enquadradas no esquema que acabamos de mencionar: o cálculo de direções de propagação de ondas em processamento de sinais ([27]), a seleção de modelos em estatística ([24]), a identificação de sistemas em engenharia ([36]), a formação de imagens de ressonância magnética ([26]), a modelagem de problemas econômicos ([4]), a filtragem colaborativa em sistemas de recomendação ([84]) e muitas outras ([11]).

Dentre elas, se destaca o problema da separação do plano de fundo (*background/foreground separation*) no contexto de sistemas de videovigilância. Nesse caso, a sequência de imagens do plano de fundo, o qual se supõe variar pouco, é modelada como um subespaço de dimensão reduzida, e os objetos em movimento são atrelados ao termo esparsos.

Podemos formular tal problema em termos matemáticos da seguinte maneira: dada uma matriz  $A \in \mathbb{R}^{m \times n}$  que é a soma de duas componentes  $S^*$  e  $L^*$  de mesmas dimensões, sendo  $S^*$  esparsa e  $L^*$  de posto baixo, queremos recuperar ambas as componentes resolvendo o problema de otimização

$$\begin{aligned} & \underset{S, L}{\text{minimizar}} && \gamma \text{nnz}(S) + \text{rank}(L) \\ & \text{sujeito a} && S + L = A, \\ & && S \in \mathcal{A}_1, L \in \mathcal{A}_2, \end{aligned} \tag{1.1}$$

em que  $\text{nnz}$  representa o número de entradas não nulas,  $\text{rank}$  é a função posto matricial,  $\gamma > 0$  é um parâmetro de compromisso que balanceia a importância que a esparsidade e o posto têm e, por último,  $\mathcal{A}_1$  e  $\mathcal{A}_2$  são subconjuntos convexos de  $\mathbb{R}^{m \times n}$ . O problema da decomposição em matrizes esparsa e de posto baixo pode ser chamado de Problema SLRMD, de *Sparse and Low-Rank Matrix Decomposition*, em inglês<sup>1</sup>.

## Caracterização e Relaxação Convexa

O Problema (1.1) tem natureza combinatória e é classificado como NP-difícil<sup>2</sup> na teoria de complexidade computacional, o que, na prática, torna muito improvável a existência de um algoritmo para resolvê-lo em tempo polinomial, no caso geral (cf. [13, p. 16]). No entanto, é comum que sejam adotadas relaxações convexas na função objetivo com o intuito de viabilizar sua resolução computacional.

Tipicamente, as funções convexas adotadas em substituição a  $\text{nnz}$  e  $\text{rank}$  são a norma 1 vetorial, que corresponde à soma dos valores absolutos das entradas do vetor, e a norma nuclear, que se trata da soma dos valores singulares matriciais. Portanto, considerando ainda a simplificação  $\mathcal{A}_1 = \mathcal{A}_2 = \mathbb{R}^{m \times n}$ , passamos a ter o seguinte problema convexo e não diferenciável em  $2mn$  variáveis e com  $mn$  restrições de igualdade:

$$\begin{aligned} \underset{S, L \in \mathbb{R}^{m \times n}}{\text{minimizar}} \quad & \Phi(S, L) \stackrel{\text{def}}{=} \gamma \|\text{vec}(S)\|_1 + \|L\|_* \\ \text{sujeito a} \quad & S + L = A, \end{aligned} \tag{1.2}$$

em que  $\text{vec}$  transforma a matriz  $S$  em um vetor empilhando suas colunas.

## Recuperabilidade da Solução

As condições de recuperabilidade das soluções  $S^*$  e  $L^*$  para o Problema (1.2) foram estudadas em ao menos dois trabalhos, em circunstâncias distintas: o de Chandrasekaran *et al.* ([25]), motivado por aplicações em identificação de sistemas e aprendizagem de modelos probabilísticos gráficos, e o de Candès *et al.* ([21]), em instâncias de médio e grande porte que, além do erro capturável por  $S$ , podem ter algumas entradas indisponíveis, numa generalização do problema de *Matrix Completion* ([22]).

Chandrasekaran *et al.* definem em [25] duas quantidades determinísticas relacionadas a  $S^*$  e  $L^*$ , e provam que, se o produto entre elas for *pequeno*, a recuperação é exata para *certa faixa de valores* do parâmetro  $\gamma$ , a qual depende dos dados e cujo cálculo pode requerer a resolução de outros problemas de otimização convexa. Por

<sup>1</sup> Todas as siglas adotadas neste trabalho que venham acompanhadas de uma termo explicativo em itálico têm sua origem na língua inglesa, razão pela qual passamos a omitir o idioma de origem.

<sup>2</sup> NP vem de *Non-deterministic Polynomial-time*.

sua vez, Candès *et al.* demonstram em [21] que, para  $\gamma$  assumindo o valor universal  $1/\sqrt{\max\{m, n\}}$ , se  $L^*$  tem vetores singulares (i.e., componentes principais) *razoavelmente espalhados*, então a recuperação é exata com *alta probabilidade*, independentemente dos padrões de esparsidade de  $S^*$  (como a magnitude ou os sinais das entradas), com a única hipótese de que a distribuição da posição das entradas não nulas seja aleatória. Note que há um certo contraponto entre as condições: uma é determinística, porém não fornece um valor universal para  $\gamma$ , enquanto outra faz uma suposição sobre a localização dos elementos de  $S$  que não condiz com várias aplicações, porém provê um valor para  $\gamma$  e engloba o caso em que há entradas matriciais indisponíveis.

## Reformulação SDP e Métodos de Primeira Ordem

O Problema (1.2) pode ser reformulado como um problema de Programação Semidefinida ([95]) e resolvido com eficiência por meio de métodos de pontos interiores ([44, 45, 93, 94]), como sugerido em [25]. No entanto, a eficiência dos métodos comuns de pontos interiores é seriamente comprometida em problemas de grande porte devido à complexidade [14, p. 24]  $\mathcal{O}\left((m n \min\{m, n\})^2\right)$  do cálculo da direção do passo, no caso de uma matriz de dados  $m \times n$ . Uma consequência da reformulação é o aumento da dimensão, que passa de  $m \times n$  para  $(m + n) \times (m + 2)$  ([25, Apêndice A]), e torna seu uso proibitivo em problemas grandes.

A dificuldade dos métodos de pontos interiores com problemas de grande porte causou um intenso desenvolvimento de métodos de primeira ordem, a classe mais apropriada para a minimização das normas 1 e nuclear em problemas com muitas variáveis. Dentre eles, se destacam na resolução do nosso problema os métodos do tipo Lagrangiano aumentado [9, 38] e métodos de gradiente proximal<sup>3</sup> [28, 87]. O impulsionamento dos métodos para minimização da norma nuclear tem suas origens na minimização da norma 1 e suas aplicações em *compressed sensing* ([5, 20, 48, 102]). Cai *et al.* [19] mostraram que o chamado *Singular Value Thresholding* pode ser usado para minimização do posto em *matrix completion* ([22, 84]).

Alguns algoritmos conciliaram uma iteração de ponto fixo com os chamados operadores de retração ou limiarização branda (*shrinkage / soft thresholding*), tirando proveito da separabilidade das funções objetivos que são dadas pela soma de duas outras funções (*composite functions*) em que, tipicamente, uma delas tem boas propriedades de suavidade e outra, de convexidade – como é o caso no Problema (1.2), por exemplo. Além disso, alguns métodos passaram a incorporar estratégias de continuação homotópica sobre o valor do parâmetro  $\gamma$ , de forma a diminuir-lo sucessivamente a fim de obter melhores resultados de convergência. Como exemplos, temos a minimização da norma 1 nos contextos de *compressed sensing* ([5, 20, 48, 102]) e a da norma nuclear, em

<sup>3</sup> Chamados também de *forward-backward splitting methods*, dentre outras denominações.

*matrix completion* ([22, 84]). Devido às semelhanças, logo as estratégias de otimização foram estendidas à combinação linear dessas normas, com o nome de Análise de Componentes Principais Robusta (RPCA) ([21, 100]).

A busca por métodos computacionais capazes de lidar com os problemas de grande porte conduziu aos algoritmos do tipo *Iterative Thresholding* (Limiarização Iterativa, em tradução livre) para minimização da norma 1, como em [20, 102], e as versões do tipo *Singular Value Thresholding* (Limiarização de Valores Singulares, também em tradução direta) para minimização da norma nuclear, como vemos em [19].

Em [74], Nesterov generalizou para otimização não suave as ideias de um de seus trabalhos anteriores sobre algoritmos de primeira ordem “otimais” para problemas suaves [73], e usou-as num problema de regularização via norma 1 em [75]. Paralelamente, Beck e Teboulle [5] também estenderam as ideias presentes em [73] e aplicaram-nas em problemas convexos não suaves, demonstrando a taxa de convergência “ótima” de  $O(1/k^2)$  de seu método de primeira ordem chamado FISTA [5], uma versão acelerada de seus antecessores da classe ISTA [29] para problemas lineares inversos estudados no contexto de processamento de sinais e imagens. Segundo os autores, a demonstração é válida para funções objetivo que são a soma de uma função suave convexa com um termo regularizador convexo não necessariamente suave.

Diversos algoritmos baseados na construção do Lagrangiano aumentado foram publicados como tentativas de obter soluções mais rapidamente do que os preexistentes. Frequentemente, esses métodos eram associados à sigla ALM, de *Augmented Lagrangian Method*. Lin et al. [62] propuseram duas versões para o ALM: o ALM Exato (EALM) e sua versão melhorada, o ALM Inexato (IALM), o qual converge praticamente tão rápido quanto o anterior porém com número de SVDs parciais significativamente menor. Ainda assim, o IALM não se aproveita da estrutura separável do problema, tanto na função objetivo quanto nas restrições, e otimiza simultaneamente em  $S$  e em  $L$ . O avanço nesse sentido veio com o Método dos Multiplicadores com Direções Alternadas (ADMM), presente nos trabalhos de Yuan e Yang [103] e Candès et al. [21].

Tanto o ADMM e o FISTA quanto o método proposto neste trabalho podem ser conectados a uma classe de métodos chamada “Métodos para Problemas Monótonos” ([32, Capítulo 12]), a qual engloba os Métodos de Projeção (*Projection Methods*), os Métodos de Ponto Proximal (*Proximal Point Methods*), os Métodos de Separação (*Splitting Methods*) entre outros, na nomenclatura de Facchinei e Pang em [32]. O livro [32], indicado por Beck e Teboulle como uma introdução ao assunto, inclui diversos resultados de convergência.

Nos últimos anos, tem sido intenso o ritmo de desenvolvimento de pesquisa que, direta ou indiretamente, abordam o Problema (1.2). Há dezenas de variações de cada tipo de método e suas descrições aqui seriam pouco proveitosas. Direcionamos

os leitores às seguintes revisões: [2, 13, 15].

## 1.2 RPCA e VCA

Na área de Análise de Dados em Estatística e, de forma mais abrangente, no campo multidisciplinar da Ciência de Dados, o Problema (1.2) está associado à chamada Análise de Componentes Principais Robusta (RPCA, de *Robust Principal Component Analysis*) e recebe o nome de PCP (*Principal Component Pursuit*). Nesse contexto, a matriz  $L$  também está vinculada à busca por um subespaço de dimensão reduzida que seja capaz de capturar uma suposta essência intrínseca dos dados, os quais frequentemente estão em espaços de dimensões enormes, da ordem de milhões ou até mesmo bilhões de variáveis. Por sua vez, a matriz  $S$  foi pensada para acomodar ruídos amorfos (não estruturados) e anomalias inerentes à geração dos dados, já que não há limitações para a magnitude de suas entradas e nem pressuposições sobre a localização desses elementos não nulos na matriz. De fato, o papel desempenhado por  $S$  nessas condições é o que justifica a robustez do RPCA e, essencialmente, o diferencia do clássico PCA (*Principal Component Analysis*, ou Análise de Componentes Principais<sup>4</sup>), uma das ferramentas mais antigas de análise multivariada ([52, 57, 80]).

A separação do plano de fundo é também uma tarefa básica da chamada Análise de Conteúdo de Vídeos (VCA, de *Video Content Analysis*), dentro da área da Visão Computacional, e compõe os primeiros passos de detecção de movimento em sistemas de videovigilância, os quais podem envolver posteriormente ações de rastreamento e reconhecimento ([12]). Sua aplicação pode ocorrer em diferentes partes do processo, principalmente na criação de um modelo inicial para o plano de fundo (*model initialization*), mas também na manutenção desse modelo (*model maintenance*) e na extração dos objetos em movimento (*foreground detection*), dentre outras ([91]).

Apesar da importância prática do assunto, destacamos que *não* é um objetivo deste trabalho aprimorar soluções tecnológicas para sistemas de videovigilância, mas sim ilustrar, na prática e com problemas de porte relevante, a potencialidade e as limitações dos métodos e modelos computacionais estudados.

## 1.3 Nossa abordagem para o problema

Neste trabalho, investigamos a decomposição matricial em componentes esparsa e de posto baixo, com experimentos computacionais envolvendo não apenas protótipos artificiais aleatórios mas também instâncias com filmagens reais que

<sup>4</sup> Outras terminologias podem ser usadas para a mesma técnica, dependendo da área de estudo, como: *Empirical Orthogonal Functions*; *Factor Analysis*; *Eigenvector Analysis*; *Latent Vector Analysis*; etc. ([57, p. v]).

resultam em problemas de porte da ordem de dezenas de milhões de variáveis de otimização. Construímos um método numérico do tipo gradiente proximal com passo espectral que está inserido numa estrutura de continuação homotópica, a qual opera sobre o parâmetro de suavização da função objetivo.

### 1.3.1 Remodelagem e Resolução Numérica

Propomos a suavização do termo da norma 1 em (1.2) usando as técnicas de Nesterov ([74]), as quais sob certas condições resultam na função suave  $h_\mu$  de Huber<sup>5</sup>, em que  $\mu$  é o parâmetro de suavização. Como veremos na Seção 2.7,  $h_\mu$  é uma subaproximação para a norma 1 e tem gradiente Lipschitz contínuo com constante igual a  $\mu$ . Além da suavização em parte da função objetivo, impomos factibilidade por meio da substituição  $S = A - L$  e obtemos, assim, a aproximação  $\Phi_\mu$ :

$$\Phi_\mu(L) \stackrel{\text{def}}{=} \gamma h_\mu(A - L) + \|L\|_* \lesssim \Phi(A - L, L), \quad (1.3)$$

em que  $\lesssim$  indica a subaproximação. Com isto, minimizamos  $\Phi_\mu$  iterativamente construindo, a cada iteração  $k$ , um modelo quadrático local  $q_k$  com o gradiente de  $h_\mu$  e com uma matriz Hessiana escalar obtida inicialmente a partir do tamanho do passo espectral de Barzilai-Borwein-Raydan [3, 83], de forma que

$$\Phi_\mu(L^{(k)}) \approx Q_k(L^{(k)}) \stackrel{\text{def}}{=} \gamma q_k(A - L^{(k)}) + \|L^{(k)}\|_*. \quad (1.4)$$

Conforme veremos na Seção 3.2, a minimização do modelo  $Q_k$  tem solução fechada que pode ser expressa como a aplicação do operador proximal da norma nuclear sobre um passo tentativo na direção de máxima descida para a função de Huber. Como é comum na adoção do passo espectral, incorporamos uma busca linear com critério não monótono.

A determinação do valor de  $\mu$  não é uma tarefa trivial, já que valores grandes conduzem a incorreções na aproximação e valores pequenos podem acarretar instabilidade numérica. Nossos experimentos mostram que uma estratégia promissora é resolver uma sucessão de problemas em que  $\Phi_{\mu_j}$  é minimizada, com  $(\mu_j)$  sendo uma sequência decrescente que se aproxima de zero, aproveitando o ponto final de cada etapa na subsequente, num esquema denominado *continuação homotópica*. Agregando todas essas ideias, construímos o que vamos chamar de Método do Gradiente Proximal Espectral Homotópico (SPGM, de *Spectral Proximal Gradient Method*).

<sup>5</sup> A função de Huber ( $h$ ), tipicamente definida em termos de uma variável real, é adotada em estatística como uma função de perda ou penalização por ser menos suscetível a *outliers* em processos de regressão. Sua versão multivariável ( $h_\mu$ ) será explicitada na Seção 2.7, Equação (2.71).

### 1.3.2 Métodos Semelhantes

O SPGM se assemelha a diversos outros métodos por pertencer à classe dos métodos de gradiente proximal ou, ainda, por compartilhar outras características não intrínsecas à classe. No entanto, até onde vai nosso conhecimento, não há outro método que seja parecido o suficiente a ponto de nomear o que construímos neste trabalho. Dentre os mais semelhantes, destacamos alguns a seguir.

Apesar de vários métodos estarem equipados com alguma estratégia de continuação homotópica, eles tipicamente operam no parâmetro de compromisso entre a esparsidade e o posto, na função objetivo, e não no parâmetro de suavização, como faz o SPGM. A exceção é o PSPG, de [2], que adota as mesmas estratégias de suavização, as ideias dos operadores proximais e a homotopia sobre o parâmetro de suavização. No entanto, os autores reduzem o valor desse parâmetro a cada iteração, e só até certo ponto, mantendo-o fixo de determinado momento em diante. Além disso, não contam com passo de busca linear e, conseqüentemente, não aderem à filosofia Barzilai-Borwein para o passo espectral.

O SPGM também é parecido com o método proposto por [53] e com o próprio SpaRSA ([101]). A diferença é que, em ambos, a homotopia ocorre sobre  $\gamma$ , e não  $\mu$ . Além disso, a direção do passo é escolhida combinando-se os dois últimos iterandos, e não apenas o atual, como o SPGM faz.

O FISTA, de Beck e Teboulle ([5]), além de não ter continuação homotópica, também se apoia nos dois últimos iterandos para determinar o seguinte, usando uma combinação linear bastante específica. Já o SPGM aplica o operador proximal no ponto mais recente, assim como faz o ISTA, precursor do FISTA. Por fim, temos o FCP [66], que também realiza iterações homotópicas no parâmetro  $\gamma$ , dentre outras diferenças ainda mais substanciais.

### 1.3.3 Metodologia de comparação

Estabelecemos como referência comparativa um dos métodos estado-da-arte para a resolução do problema PCP (1.2), o *Alternating Direction Method of Multipliers* (ADMM), em implementação própria. Trata-se de um dos métodos mais adotados na literatura ([14, 27, 65, 90, 97]), utilizado nos artigos de Candès *et al.* ([21]) e Yuan e Yang ([103]) que, em certo sentido, norteiam os pesquisadores em RPCA. Além disso, os resultados reportados em [21] mostram que o ADMM é capaz de resolver problemas de milhões de variáveis com cerca de vinte iterações, se mostrando surpreendentemente pouco sensível ao aumento de dimensão.

### 1.3.4 Contribuições deste trabalho

São duas as principais contribuições deste trabalho ([72]) na resolução do Problema (1.2) associado ao contexto da decomposição de imagens. A primeira é a elaboração de um novo método computacional de otimização que acopla um algoritmo de gradiente proximal com tamanho de passo do tipo Barzilai-Borwein a um esquema de continuação homotópica que opera no parâmetro de suavização do termo da norma 1 da função objetivo, resolvendo assim uma sequência de problemas irrestritos que converge para o problema convexo relaxado original. Esse método, que combina estratégias preexistentes, mostra-se robusto e mais eficiente do que o método dos multiplicadores com direções alternadas na resolução de problemas com filmagens reais, segundo a evidência experimental de que a precisão de 0.1% de erro relativo é suficiente para recuperar as imagens digitais que são solução do problema de otimização. Com esses resultados, torna-se promissora a alternativa de se suavizar o termo da norma nuclear da função objetivo.

A outra contribuição é a proposição de uma heurística simples para construção do ponto inicial, segundo a qual se somam sucessivamente as matrizes de posto 1 advindas da decomposição em valores singulares da matriz de dados até que o valor da função objetivo convexa relaxada deixe de decrescer. Trata-se de um procedimento atrelado ao Problema (1.2), e não a um determinado método. Como consequência dessa estratégia, observa-se uma aceleração considerável do ADMM na resolução dos problemas de videovigilância.

## 1.4 Algumas Notações

As unidades constituintes de uma matriz serão chamadas de entradas ou elementos, ficando o termo *matriz-componente* – ou, simplesmente, (a) *componente* – reservado para se referir a cada uma das parcelas matriciais relacionadas à decomposição da matriz do problema. A componente candidata à matriz esparsa da decomposição será denotada por “ $S$ ” e, por concisão, será chamada também de *componente de esparsidade*. De forma análoga, a matriz candidata à parcela de posto baixo será representada por “ $L$ ” e denominada *componente de posto*. Podemos fazer menção à matriz resultante da justaposição de ambas componentes e designá-la por “[ $S L$ ]”. Em certos momentos, será conveniente interpretar a junção de componentes como um par ordenado de matrizes, isto é,  $(S, L)$ .

As dimensões das matrizes-componentes são sempre representadas pelas letras minúsculas  $m$  e  $n$ , ou apenas  $n$ , em caso de matrizes quadradas. Dessa forma, por exemplo, se  $S$  e  $L$  têm  $m$  linhas e  $n$  colunas cada, isto é, dimensão  $m \times n$ , então a matriz [ $S L$ ] tem dimensão  $m \times 2n$ .

Frequentemente, no decorrer do trabalho, será necessário realizar a conversão de matrizes em vetores coluna, pelo que é conveniente definir um operador que fixe definitivamente a reestruturação dos elementos matriciais e facilite a notação.

**Definição 1** (Vetorização). Dada uma matriz  $C = (c_{ij})$  de dimensão  $m \times n$ , definimos como vetorização o resultado da reorganização posicional dos elementos de  $C$  que a converte em um vetor coluna de dimensão  $m n \times 1$ , denotado por  $\text{vec}(C)$ , empilhando sucessivamente as colunas de  $C$  uma em cima da outra. Com isto, a correspondência entre as entradas de  $C$  e os elementos de  $v = \text{vec}(C)$  é dada pela chamada indexação linear, ou seja,

$$v_\ell = c_{ij}, \quad (1.5)$$

em que

$$i = i(\ell) = 1 + \text{mod}(\ell - 1, m),$$

$$j = j(\ell) = 1 + [(\ell - 1) - \text{mod}(\ell - 1, m)]/m,$$

sendo  $\text{mod}(\ell, m)$  o resto da divisão inteira de  $\ell$  por  $m$ . Na notação do MATLAB, tais conversões podem ser feitas diretamente por meio dos comandos  $v = C(:)$  e  $C = \text{reshape}(v, m, n)$ .

Representaremos a quantidade de entradas não nulas de uma matriz  $C$  por  $\text{nnz}(C)$ , do inglês *number of nonzero elements*, e o conjunto suporte de  $C$ , por  $\text{supp}(C)$ .

Chamaremos de taxa de esparsidade – *spr* (do inglês *sparsity rate*) – a porcentagem que expressa a razão entre o número de elementos não nulos de uma matriz e a quantidade total de seus elementos, arredondada para algumas casas decimais quando necessário. Observe que o termo *esparsidade* é utilizado aqui para se referir ao preenchimento da matriz por elementos não nulos. Se bem consideramos *densidade* um termo mais apropriado, optamos por manter a nomenclatura que lhe é dada na literatura sobre o tema.

O posto de uma matriz é representado por seu termo em inglês, *rank*, ou *rnk*, na forma abreviada. Em diversas situações, é mais conveniente falar do que chamaremos de posto percentual – *rkr* (de *rank rate*, também da língua inglesa) –, que é o percentual do posto com relação à menor dimensão da matriz em questão, o qual também pode ser arredondado a certo número de casas decimais quando necessário.

Na representação numérica, adotaremos o ponto (“.”) como separador decimal, no lugar da vírgula, que será reservada para separar elementos em listas, vetores e etc. O formato da notação científica adotado aqui terá seus dois fatores multiplicativos (a mantissa e a potência de dez) separados pelo símbolo “×”. Na parte experimental, usamos também a notação do MATLAB que explicita apenas a

mantissa e o expoente, demarcando sua separação por meio da letra “e” ou “E”. Com isso, a título de ilustração, temos as seguintes representações equivalentes: 0.000123,  $1.23 \times 10^{-4}$  e 1.23e-04.

## 1.5 Organização do Trabalho

O restante deste trabalho está organizado da seguinte maneira. Na primeira parte da tese, no Capítulo 2, apresentamos os tópicos sobre a teoria das condições de otimalidade para o Problema (1.2) a partir de elementos de dualidade, assim como desenvolvemos as suavizações para as normas 1 e nuclear. A descrição do ADMM e do SPGM são feitas no Capítulo 3, em que são fornecidos os pseudocódigos de cada um desses métodos. A segunda parte do trabalho, que trata dos experimentos numéricos, é iniciada no Capítulo 4 com os testes computacionais envolvendo as instâncias aleatórias, desde sua construção e caracterização até a comparação de desempenho dos dois métodos nesses problemas. O Capítulo 5 é a preparação para os experimentos com os vídeos; nele são conceituados o que chamamos de Espaço de Exibição, onde são definidas medidas adicionais de comparação, e também as Soluções de Referência, além de ser estabelecida a precisão alvo que entendemos ser condizente com a aplicação. O conjunto de problemas-testes vindos de filmagens reais é apresentado no Capítulo 6, junto com suas Soluções de Referência, e tem suas características (entradas e valores singulares) comparadas às dos problemas artificiais. O Capítulo 7 é onde estão os principais resultados deste trabalho, os quais se tratam da comparação de desempenho entre o SPGM e o ADMM nos problemas de videovigilância. Por sua vez, o Capítulo 8 descreve a heurística de *warm-start* para obtenção dos pontos iniciais e seu impacto nos resultados numéricos, além de exibi-los e compará-los visualmente. Finalmente, o Capítulo 9 traz as considerações finais com algumas conclusões e trabalhos futuros, e é procedido pelos Apêndices e Anexos contendo dados relevantes porém não centrais nas análises e conclusões feitas a partir dos demais capítulos.

## Parte II

# Teoria e Métodos Computacionais

## 2 FUNDAMENTOS TEÓRICOS

Neste capítulo, reintroduzimos o Problema da Decomposição de Matrizes em Componentes Esparsa e de Posto Baixo, e apresentamos alguns resultados teóricos já publicados a respeito desse problema e dos métodos computacionais que utilizaremos para resolvê-lo.

Nosso objetivo neste capítulo é reunir resultados relevantes para o entendimento do problema, algumas de suas características e limitações, além de direcionar a leitura para publicações de relevância sobre os tópicos abordados. Dentre outros assuntos, tratamos da existência e unicidade de soluções, da construção do problema dual Lagrangiano e das condições de otimalidade via conjunto subdiferencial.

Assumimos que o(a) leitor(a) tem familiaridade com os conteúdos básicos de álgebra linear, cálculo, cálculo numérico e otimização. Em caso de dúvidas, recomendamos que consulte os livros e as publicações extensivamente referenciados ao longo do texto.

### Sobre funções próprias

Em otimização convexa, tipicamente lidamos com funções de variáveis reais a valores no conjunto real estendido, isto é,  $f : \mathcal{D} \rightarrow [-\infty, +\infty]$ , em que se convencionou chamar de *domínio efetivo* o conjunto  $\text{dom}(f) \stackrel{\text{def}}{=} \{x \in \mathcal{D} \mid f(x) < \infty\}$ . Isto porque, mesmo nas operações mais fundamentais com funções reais, como a transformação de conjugação, é comum que a função resultante tenha características que justificam a atribuição de “ $\infty$ ” a seu valor – uma dentre várias situações em que a notação é vantajosa. Em contrapartida, muitos resultados precisam de mais cuidado ao serem enunciados e interpretados. Ao longo deste e dos demais capítulos, a menos que se explicitar o contrário, as funções que abordamos são *funções próprias*, o que significa que assumem um valor finito em pelo menos um ponto de seu domínio, ou seja,  $\text{dom}(f) \neq \emptyset$ , e, adicionalmente,  $f(x) > -\infty$  para todo  $x$  em  $\mathcal{D}$ .

### 2.1 Formulações Matemáticas

Suponha que dispomos de uma matriz  $A \in \mathbb{R}^{m \times n}$  que é resultado da soma de uma matriz  $S^*$  esparsa com outra matriz  $L^*$  de posto baixo, ambas desconhecidas. Queremos recuperar as componentes  $S^*$  e  $L^*$  apenas a partir dessas informações. Uma maneira de formular esse problema matematicamente é por meio da minimização do posto e da esparsidade apropriados, combinados numa função objetivo de um

problema de otimização com restrições de igualdade: dada  $A \in \mathbb{R}^{m \times n}$  tal que

$$A = S^* + L^*,$$

em que  $S^*$  e  $L^*$  são as componentes esparsa e de posto baixo, respectivamente, queremos recuperar  $S^*$  e  $L^*$  como soluções do problema de otimização

$$\begin{aligned} & \underset{S, L \in \mathbb{R}^{m \times n}}{\text{minimizar}} && \gamma \text{nnz}(S) + \text{rank}(L) \\ & \text{sujeito a} && S + L = A, \\ & && S \in \mathcal{A}_1, L \in \mathcal{A}_2, \end{aligned} \tag{2.1}$$

em que  $\gamma \in (0, \infty)$  é um parâmetro de balanceamento entre as grandezas em disputa e  $\mathcal{A}_1$  e  $\mathcal{A}_2$  são subconjuntos convexos de  $\mathbb{R}^{m \times n}$ .

Da maneira como foi formulado, o Problema (2.1) é mal-posto<sup>1</sup> e intratável devido à sua natureza combinatória ([59]). No entanto, podemos considerar uma estratégia bastante conhecida de relaxação convexa que consiste em adotar a norma 1 vetorial no lugar da função que retorna o número de elementos não nulos da matriz, a qual chamaremos pela sigla em inglês: *nnz*, de *number of nonzero elements*. Analogamente, podemos substituir a função posto matricial pela chamada norma nuclear<sup>2</sup>, que consiste na soma dos valores singulares da matriz em que opera.

A razão por que faz sentido adotar essas relaxações convexas está associada aos envelopes convexos das funções cardinalidade e posto. Consideremos um conjunto  $C$  convexo e uma função  $f : C \rightarrow \mathbb{R}$  não necessariamente convexa. O envelope convexo (ou envoltória convexa) da função  $f$  é definido como a maior função convexa  $g$  tal que  $g(x) \leq f(x)$  para todo  $x$  em  $C$ . Isso significa que  $g$  é a melhor subaproximação convexa ponto a ponto para  $f$ . Ocorre que, surpreendentemente, os envelopes convexos da cardinalidade em  $\{x \in \mathbb{R}^n \mid \|x\|_\infty \leq 1\}$  e do posto em  $\{X \in \mathbb{R}^{m \times n} \mid \|X\|_2 \leq 1\}$  são precisamente as normas 1 e nuclear. Podemos encontrar uma demonstração para o caso da norma nuclear em [35], sendo o da norma 1 um caso particular.

Neste trabalho, para invocar a soma dos valores absolutos das entradas de uma matriz, usamos a norma 1 vetorial aplicada ao operador *vec* que rearranja os elementos da matriz como um vetor coluna. Para  $X = (X_{ij}) \in \mathbb{R}^{m \times n}$ , temos:

$$\|\text{vec}(X)\|_1 \stackrel{\text{def}}{=} \sum_{i=1}^m \sum_{j=1}^n |X_{ij}|. \tag{2.2}$$

<sup>1</sup> No sentido de Hadamard [47], um problema é bem posto se sua solução existe, é única e varia continuamente com os dados do problema.

<sup>2</sup> A norma nuclear também aparece na literatura como norma-1 de Schatten ([76]), norma- $n$  de Ky-Fan (cf. [33, 99] e [51, Seção 7.4]) e, ainda, como norma da classe do traço ([51, Seção 5.2]).

Por sua vez, a norma nuclear de  $X$ , representada por  $\|X\|_*$ , corresponde à soma de seus valores singulares. Ou seja,

$$\|(X)\|_* \stackrel{\text{def}}{=} \sum_{i=1}^{\text{rank}(X)} \sigma_i(X) = \sum_{i=1}^n \sqrt{\lambda_i(X^\top X)} = \text{tr} \left( \sqrt{X^\top X} \right), \quad (2.3)$$

em que  $\sigma_i(X)$  é o  $i$ -ésimo valor singular de  $X$ ,  $\lambda_i(X^\top X)$  é um autovalor de  $X^\top X$ ,  $\text{tr}$  é a função traço matricial e  $\sqrt{X^\top X}$  é a raiz quadrada<sup>3</sup> da matriz semidefinida positiva  $X^\top X$ . Para que não haja confusão com a normas nuclear, relembramos que uma outra maneira de representar a norma de Frobenius é

$$\|X\|_F \stackrel{\text{def}}{=} \sqrt{\sum_{i=1}^m \sum_{j=1}^n |X_{ij}|^2} = \sqrt{\sum \sigma_i^2} = \sqrt{\text{tr}(X^\top X)}. \quad (2.4)$$

Usando a notação apresentada anteriormente e assumindo, por simplicidade, que  $\mathcal{A}_1$  e  $\mathcal{A}_2$  são o próprio  $\mathbb{R}^{m \times n}$ , o problema convexo original pode ser reescrito como:

$$\begin{aligned} & \underset{S, L \in \mathbb{R}^{m \times n}}{\text{minimizar}} \quad \left[ \Phi(S, L) \stackrel{\text{def}}{=} \gamma \|S\|_1 + \|L\|_* \right] \\ & \text{sujeito a} \quad S + L = A. \end{aligned} \quad (2.5)$$

Por questões de estabilidade numérica nos experimentos computacionais, realizamos também a mudança

$$\gamma = \frac{t}{1-t}, \quad (2.6)$$

para  $t \in (0, 1)$ , e então multiplicamos a função objetivo por  $(1-t) \in (0, 1)$ , o que garantidamente não altera a solução do problema, embora modifique seu valor ótimo. Obtemos, por fim, o seguinte problema:

$$\begin{aligned} & \underset{S, L \in \mathbb{R}^{m \times n}}{\text{minimizar}} \quad \left[ \Psi(S, L) \stackrel{\text{def}}{=} t \|S\|_1 + (1-t) \|L\|_* \right] \\ & \text{sujeito a} \quad S + L = A. \end{aligned} \quad (2.7)$$

A distinção de nomenclatura entre  $\Phi$  e  $\Psi$  é conveniente porque a primeira será usada principalmente no desenvolvimento teórico por condensar as contas, enquanto a última será aquela a qual corresponderão os resultados numéricos da Parte III.

## 2.2 Convexidade Não Estrita

Sabemos que, sob convexidade, temos a valiosa garantia de que todo mínimo local é também mínimo global, ainda que possam existir infinitos minimizadores – ou,

<sup>3</sup> Uma matriz  $M \in \mathbb{R}^{n \times n}$  semidefinida positiva é a raiz quadrada de  $N \in \mathbb{R}^{n \times n}$  se  $N = M^2$ . No caso geral em que  $M$  é uma matriz qualquer, define-se que  $M$  é a *raiz quadrada principal* de  $A$  se: (i)  $M^2 = A$ ; (ii) os autovalores de  $M$  têm parte real positiva (cf. [40, Seção 9.4.2]).

mesmo, nenhum. Sob convexidade estrita, outra excelente garantia nos é dada: a de que existe, no máximo, um minimizador global (cf. [6, Prop. 3.1.1]). No entanto, como veremos agora, este não é o caso da função objetivo do Problema (2.5)

A função objetivo  $\Phi$  (ou  $\Psi$ ) é convexa, já que se trata de uma combinação convexa de normas. Entretanto, como mostraremos a seguir, nenhuma das duas normas é estritamente convexa, pelo que podemos constatar que não estamos na situação de convexidade estrita com essa função objetivo. Sabemos que, por definição, uma função  $f : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$  é estritamente convexa se, para todo  $X$  e  $Y$  em  $\mathbb{R}^{m \times n}$  e  $\alpha \in (0, 1)$ , temos que

$$f(\alpha X + (1 - \alpha)Y) < \alpha f(X) + (1 - \alpha)f(Y) \quad \text{sempre que} \quad X \neq Y. \quad (2.8)$$

Em particular, uma norma  $\|\cdot\|$  é estritamente convexa se e somente se (cf. [104, p. 61])

$$\|x\| = \|y\| = 1, \|x + y\| = 2 \quad \text{implicar em} \quad x = y. \quad (2.9)$$

A desigualdade triangular para a norma 1 no caso unidimensional se trata da desigualdade triangular usual para o valor absoluto, o qual sabemos que não é estritamente convexo. Por outro lado, para a norma nuclear, se tomarmos duas matrizes simétricas  $X$  e  $Y$  distintas tais que  $XY = YX$  podemos mostrar que a desigualdade triangular é satisfeita na igualdade. Portanto, é possível concluir que nem  $\Phi$  nem  $\Psi$  são funções estritamente convexas.

Em suma, essas são as principais características do Problema (2.7):

- $2mn$  variáveis;
- função não linear e não suave;
- $mn$  restrições lineares de igualdade;
- convexidade não estrita.

Outras características de grande relevância no estudo de um problema são as que dizem respeito à dualidade, culminando na caracterização do problema dual Lagrangiano, como fazemos a seguir.

## 2.3 Dualidade

A fim de determinar o problema dual Lagrangiano do Problema (2.7), consideramos inicialmente a função Lagrangiana<sup>4</sup>  $\mathcal{L}$  associada, dada por  $\mathcal{L} : \mathcal{D} \rightarrow \mathbb{R}$ , com  $\mathcal{D} = \mathbb{R}^{m \times n} \times \mathbb{R}^{m \times n} \times \mathbb{R}^{m \times n}$  e

$$\begin{aligned} \mathcal{L}(S, L, Z) &\stackrel{\text{def}}{=} \Psi(S, L) + \langle Z, S + L - A \rangle \\ &= t \|\text{vec}(S)\|_1 + (1 - t) \|L\|_* + \langle Z, S + L - A \rangle. \end{aligned} \quad (2.10)$$

<sup>4</sup> Ou, simplesmente, o Lagrangiano.

A ideia básica por trás da construção do Lagrangiano é de levar as restrições para junto da função objetivo por meio de uma soma ponderada das funções que definem as restrições – no nosso caso, a função  $\hat{f}(S, L) \stackrel{\text{def}}{=} S + L - A$ . Nos pontos em que a restrição original é satisfeita, o Lagrangiano coincide com a função objetivo original  $\Psi$ . As variáveis  $Z$  introduzidas são chamadas de multiplicadores de Lagrange ou, como veremos adiante, variáveis duais.

Prosseguindo como em [56], definimos

$$\Delta \stackrel{\text{def}}{=} \left\{ Z \in \mathbb{R}^{m \times n} \mid \inf_{S, L \in \mathbb{R}^{m \times n}} \mathcal{L}(S, L, Z) > -\infty \right\} \quad (2.11)$$

e a função dual  $\phi : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$  como

$$\phi(Z) \stackrel{\text{def}}{=} \inf_{S, L \in \mathbb{R}^{m \times n}} \mathcal{L}(S, L, Z). \quad (2.12)$$

Note que, assim,  $\text{dom}(\phi) = \Delta$ . A função dual tem características importantes. Por exemplo, por ser definida pontualmente como o ínfimo de uma família de funções afim (na variável  $Z$ ), é sempre uma função côncava, mesmo quando o problema de original não é convexo. Além disso, as funções duais sempre fornecem limitantes inferiores para o valor ótimo do problema original, digamos,  $\Psi^*$ , já que, como consequência direta da definição em (2.12), temos  $\phi(Z) \leq \Psi^*$ .

No caso particular do Problema (2.7), podemos escrevê-la na seguinte forma:

$$\phi(Z) = - \left( \sup_{S, L \in \mathbb{R}^{m \times n}} -\mathcal{L}(S, L, Z) \right) \quad (2.13)$$

$$= - \left\{ \sup_{S, L \in \mathbb{R}^{m \times n}} \{-t \|\text{vec}(S)\|_1 - (1-t) \|L\|_* - \langle Z, S + L - A \rangle\} \right\} \quad (2.14)$$

$$= - \langle Z, A \rangle - \left\{ \sup_{S \in \mathbb{R}^{m \times n}} \{\langle -Z, S \rangle - t \|\text{vec}(S)\|_1\} \right\} \quad (2.15)$$

$$- \left\{ \sup_{L \in \mathbb{R}^{m \times n}} \{\langle -Z, L \rangle - (1-t) \|L\|_*\} \right\}. \quad (2.16)$$

Consideremos, agora, a definição de função conjugada convexa. Ela será útil para reescrever a função dual de  $f$  em termos da conjugada de uma outra função associada.

**Definição 2** (Função Conjugada Convexa). Seja  $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$  um função que a cada número real associa um valor na reta real estendida. Definimos a função conjugada convexa  $f^*$  da função  $f$  como

$$f^*(y) \stackrel{\text{def}}{=} \sup_{x \in \mathbb{R}^n} \{\langle y, x \rangle - f(x)\}. \quad (2.17)$$

A função conjugada convexa também é conhecida como *transformada de Fenchel*. Apresentada brevemente em [37] e mais explorada posteriormente em [85], a transformada de Fenchel pode ser vista como uma generalização da clássica transformada de Legendre.

Definamos duas funções auxiliares  $f_1$  e  $f_2$  da seguinte maneira:

$$f_1(S) \stackrel{\text{def}}{=} t \|\text{vec}(S)\|_1 \quad \text{e} \quad f_2(L) \stackrel{\text{def}}{=} (1-t) \|L\|_*. \quad (2.18)$$

Assim, a função dual  $\phi$  pode ser reescrita em termos de suas conjugadas convexas como

$$\phi(Z) = -\langle Z, A \rangle - f_1^*(-Z) - f_2^*(-Z). \quad (2.19)$$

Quando se trata de uma norma em um espaço Euclidiano, é possível mostrar que sua conjugada convexa nada mais é do que a função indicadora da bola unitária de sua norma dual [17, Exemplo 3.26], isto é,

$$f(x) = \|x\| \implies f^*(y) = \begin{cases} 0, & \text{se } \|y\|_d \leq 1 \\ \infty, & \text{caso contrário.} \end{cases} \quad (2.20)$$

Sendo assim, como o conjunto  $\Delta$  definido em (2.11) exige que  $\inf_{S,L} \mathcal{L}(S,L,Z) > -\infty$ , podemos explicitá-lo a partir do domínio efetivo da função conjugada:

$$\Delta = \{Z \mid -Z \in \text{dom}(f_1^*) \text{ e } -Z \in \text{dom}(f_2^*)\} \quad (2.21)$$

$$= \{Z \mid \|Z\|_2 \leq 1-t, \|\text{vec}(Z)\|_\infty \leq t\}, \quad (2.22)$$

em que usamos o fato de que a dual da norma nuclear é a norma espectral e a dual da norma 1 é a norma infinito ([51, Seção 5.6]). Dessa forma, o problema dual associado a (2.1) é dado por

$$\begin{aligned} & \underset{Z}{\text{maximizar}} && -\langle Z, A \rangle \\ & \text{sujeito a} && \|Z\|_2 \leq 1-t \\ & && \|\text{vec}(Z)\|_\infty \leq t. \end{aligned} \quad (2.23)$$

## 2.4 Condições de Otimalidade

Definimos a seguir o subdiferencial de uma função convexa. Seus elementos, os subgradientes, são uma generalização do vetor gradiente para funções convexas não diferenciáveis, como é o nosso caso por ora. Veremos que, assim como o gradiente no caso suave, o conjunto subdiferencial desempenha um papel central no enunciado das condições de otimalidade para problemas convexas.

**Definição 3 (Subdiferencial).** Seja  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  uma função convexa. Dizemos que o vetor  $v \in \mathbb{R}^n$  é um *subgradiente* de  $f$  em um ponto  $x$  se

$$f(y) \geq f(x) + \langle v, y - x \rangle, \quad \forall y \in \mathbb{R}^n. \quad (2.24)$$

O conjunto de todos os subgradientes de  $f$  no ponto  $x \in \mathbb{R}^n$  constitui o chamado *subdiferencial* de  $f$  nesse ponto, ou seja,

$$\partial f(x) \stackrel{\text{def}}{=} \left\{ z \in \mathbb{R}^n \mid f(y) \geq f(x) + \langle z, y - x \rangle, \quad \forall y \in \mathbb{R}^n \right\}. \quad (2.25)$$

No caso das funções a valores reais, é possível mostrar a partir da definição que o subdiferencial é um conjunto compacto (fechado e limitado) não vazio. Outra propriedade importante, embora não se aplique muito em nosso contexto, é que, se  $f$  é diferenciável em algum ponto do interior de seu domínio, então o gradiente de  $f$  é o único subgradiente nesse ponto (para uma demonstração, veja [6, p. 184]). Geometricamente, se  $v \in \partial f(x)$ , então a função afim  $g(y) = f(x) + \langle v, y - x \rangle$  subestima globalmente o valor de  $f$ , cujo gráfico se encontra, portanto, acima do gráfico de  $g$  em cada ponto do domínio. Por fim, vale mencionar também que, para uma função convexa  $f : \mathcal{D} \rightarrow \mathbb{R}$ , a derivada direcional na direção  $d \in \mathcal{D}$ , definida por

$$f'(x; d) \stackrel{\text{def}}{=} \lim_{\alpha \rightarrow 0} \frac{f(x + \alpha d) - f(x)}{\alpha} \quad (2.26)$$

sempre existe e pode ser expressa por intermédio da seguinte igualdade<sup>5</sup> envolvendo o subdiferencial

$$f'(x; d) = \sup_{y \in \partial f(x)} \langle y, d \rangle. \quad (2.27)$$

Estabelecemos agora as condições de otimalidade para o Problema (2.5), o qual é composto por uma função objetivo convexa não diferenciável e restrições lineares de igualdade. Para esse tipo de problema, enunciamos as seguintes condições necessárias e suficientes de otimalidade, um caso particular da Proposição 4.7.2 de [8].

**Proposição 1 (Condições de Otimalidade).** Sejam  $f_1, f_2 : \mathbb{R}^n \rightarrow \mathbb{R}$  duas funções convexas e  $f$  sua soma, ou seja,  $f(x) = f_1(x) + f_2(x)$ . Um vetor  $x^*$  minimiza  $f$  em  $\mathbb{R}^n$  se, e somente se, existe um vetor subgradiente  $d \in \partial f(x^*)$  tal que

$$0 \in \partial f(x^*) = \partial f_1(x^*) + \partial f_2(x^*). \quad (2.28)$$

A adição de conjuntos presente na relação expressa em (2.28) pode ser definida a partir de dois conjuntos  $C$  e  $D$  como segue:

$$C + D \stackrel{\text{def}}{=} \{c + d \mid c \in C, d \in D\}. \quad (2.29)$$

<sup>5</sup> No caso em que  $f$  pode assumir valor  $\infty$ , o lado direito da igualdade vira limitante inferior para a derivada direcional (cf. [6, p. 198] ou [86, Teo. 23.4]).

Além disso, a igualdade nessa expressão é a clássica de conjuntos e, portanto, sua demonstração envolve duas partes: a da inclusão ( $\subset$ ) e a da continência ( $\supset$ ). Veja [6, Proposição 5.4.6] para acessar a demonstração sobre um caso mais geral de subdiferencial da soma de funções.

Antes de passarmos para o subdiferencial da função objetivo, fixemos uma notação. Para valores  $x \in \mathbb{R}$ , definimos

$$\text{sign}(x) = \begin{cases} +1, & \text{se } x > 0 \\ 0, & \text{se } x = 0. \\ -1, & \text{se } x < 0 \end{cases} \quad (2.30)$$

Para vetores  $x \in \mathbb{R}^n$ , estendemos a definição anterior para que  $\text{sign}(x) \in \mathbb{R}^n$  seja o vetor cuja  $i$ -ésima componente é  $\text{sign}(x_i)$ .

Estendemos também a definição de suporte de uma função para contemplar vetores  $x = (x_1, x_2, \dots, x_n)$  quaisquer, interpretando-os como funções bijetoras que associam a posição de suas componentes às componentes em si, isto é,

$$x : i \rightarrow x(i) = x_i \quad (2.31)$$

Portanto,  $\text{supp}(x)$  é o conjunto de índices associados às componentes não nulas de  $x$ .

Há diferentes maneiras de se caracterizar o subdiferencial de uma função, a depender da expressão que a define. Consideremos as caracterizações dos conjuntos subdiferenciais das normas que compõem a função objetivo do Problema (2.5). Da Definição 3, é possível mostrar (cf. [98, Exemplo 2] e [84, 481]) que, para  $x \in \mathbb{R}^n$ , temos

$$\partial \|x\|_1 = \{\text{sign}(x) + w \mid \text{supp}(x) \cap \text{supp}(w) = \emptyset, \|w\|_\infty \leq 1\}. \quad (2.32)$$

De forma análoga, para  $X \in \mathbb{R}^{m \times n}$  com decomposição SVD dada por  $X = U\Sigma V^\top$ , temos também que

$$\partial \|X\|_* = \{UV^\top + W \mid W \in \mathbb{R}^{m \times n}, U^\top W = 0, WV = 0, \|W\|_2 \leq 1\}. \quad (2.33)$$

As descrições dos subdiferenciais  $\partial \|x\|_1$  e  $\partial \|X\|_*$  dadas por (2.32) e (2.33) têm uma semelhança que pode ser explicada da seguinte forma. Partindo da interpretação da matriz  $X$  como uma transformação de  $\mathbb{R}^n$  em  $\mathbb{R}^m$  e seu efeito na bola unitária, o termo  $UV^\top$  de  $\partial \|X\|_*$  pode ser associado à rotação, ou angulação, que  $X$  impõe da base de vetores singulares em  $\mathbb{R}^n$  para a base de vetores singulares em  $\mathbb{R}^m$ . De fato, note que  $UV^\top = U(I_{m \times n})V^\top$  é uma matriz cujos valores singulares são todos iguais a 1. Já as condições  $U^\top W = 0$ ,  $WV = 0$  indicam que  $W$  e  $X$  têm espaços linha e coluna ortogonais, o que seria correspondente à disjunção entre os suportes de  $w$  e  $x$  no caso da norma 1. Portanto, conforme destacam Recht *et al.* ([84]), podemos dizer que os

subgradientes sempre assumem a forma de “ângulo” (ou sinal) mais uma possível contração numa direção ortogonal, se a norma não é diferenciável no ponto corrente.

Uma outra maneira de descrever o conjunto (2.33) pode ser obtida através da expressão da norma nuclear como a composição de duas funções,

$$\|X\|_* = \|\sigma(X)\|_1 = (g \circ \sigma)(X),$$

em que  $g$  é a função  $g(x) = \|x\|_1$  e  $\sigma(X)$  é o vetor de valores singulares de  $X$  ordenados de maneira decrescente, como mencionamos na introdução deste trabalho. De acordo com Watson ([98]),  $g$  é uma função do tipo *calibre simétrica* e, portanto, sua composição com  $\sigma(X)$  produz uma norma unitariamente invariante cujo subdiferencial pode ser expresso como

$$\partial \|X\|_* = \{UDV^\top \mid X = U\Sigma V^\top, D = \text{diag}(d), d \in \partial \|\sigma(X)\|_1\}. \quad (2.34)$$

Essa descrição é consequência dos resultados [30, Corolário 5.2] e [98, Teorema 2].

Um dos trabalhos que podem ser diretamente relacionados à descrição do subdiferencial das normas unitariamente invariantes, como é o caso da norma nuclear, é o de von Neumann [96], no qual um dos resultados fundamentais foi a conclusão de que normas matriciais unitariamente invariantes sempre podem ser escritas como a composição de uma função calibre simétrica com o vetor de valores singulares.

Lewis [60] faz uma breve, reverente e entusiástica descrição dos passos que von Neumann tomou para provar que a composição de uma função calibre simétrica com o vetor de valores singulares é, de fato, uma norma. Uma das etapas da demonstração consiste em provar a desigualdade fundamental  $\langle X, Y \rangle \leq \langle \sigma(X), \sigma(Y) \rangle$ , para toda matriz  $X$  e  $Y$ . E, segundo Lewis, essa etapa é realizada justamente considerando as condições de otimalidade para o problema de maximizar o produto interno  $\langle Y, Z \rangle$  em que  $Z$  assume valores sobre um conjunto com certa estrutura fixada. Ainda segundo Lewis, um estudo cuidadoso sobre essa desigualdade conduz à expressão para o subdiferencial das normas unitariamente invariantes (de onde podemos extrair a descrição):

$$\partial(g \circ \sigma)(X) = \{U \text{Diag}(y) V^\top \mid U, V \in \mathcal{U}^n, y \in \partial g(x), X = U \text{Diag}(x) V^\top\}, \quad (2.35)$$

para qualquer função calibre simétrica  $g$ , em que  $\mathcal{U}^n$  é o grupo de matrizes unitárias.

Vale mencionar também que a própria demonstração sobre os envelopes convexos da cardinalidade e do posto pode ser feita a partir da desigualdade do traço, de von Neumann, como comentam Recht *et al.* em [84, p. 479], a respeito da demonstração que consta em [35].

Munidos de duas maneiras de caracterizar o subdiferencial de nosso problema, passamos agora para as condições suficientes de otimalidade publicadas

por [24]. Como veremos, as condições propostas pelos autores exigem um pouco mais do que o necessário para a existência de solução, já que também estão interessados na unicidade da mesma.

## 2.5 Condições suficientes de otimalidade (Chandrasekaran *et al.*)

Os dois principais trabalhos sobre as condições de recuperabilidade ou identificabilidade da(s) solução(ões) do Problema (2.5) são os de Chandrasekaran *et al.* ([25]) e o de Candès *et al.* ([21]). Embora haja semelhança considerável entre eles, os contextos de aplicação que os motivaram são distintos.

A recuperabilidade estabelecida em [21] assume que as entradas não nulas da componente  $S^*$  da solução têm distribuição uniforme e, a partir disso, conseguem demonstrar a recuperabilidade exata inclusive fornecendo um valor universal para a constante  $\gamma$  que funcionaria em todos casos, ainda que houvesse outros valores que conduzissem a resultados melhores, do ponto de vista do desempenho do método de otimização. Por sua vez, a teoria desenvolvida em [25] não faz suposições sobre a distribuição das entradas da matriz  $L^*$  e, por isso, precisa acrescentar hipóteses aos seus teoremas de forma a evitar, por exemplo, componentes esparsas que tenham suas entradas concentradas em uma mesma linha ou coluna. Além disso, os autores estabelecem que a recuperação exata ocorre para uma faixa de valores do parâmetro  $\gamma$  que depende não apenas dos dados de entrada do problema, mas também da solução. Esse cálculo pode envolver até mesmo a resolução de outros problemas de otimização.

No que segue, retomaremos os resultados de Chandrasekaran *et al.* a partir da sua proposição final até chegar às condições de otimalidade que a originaram.

A formulação do problema da decomposição posto-esparsidade em [25] é análoga à que apresentamos no início do capítulo. Inicialmente, os autores afirmam que há dois possíveis problemas de *identificabilidade* da solução: um quando a própria matriz associada ao posto é esparsa, e o outro, quando ocorre a concentração de elementos não nulos numa mesma coluna da matriz esparsa. Assim, são definidas duas medidas que indicam se o par de matrizes escapa desses problemas e pode ser recuperado com exatidão. São elas:

$$\xi(M) \stackrel{\text{def}}{=} \max_{\substack{N \in T(M), \\ \|N\|_2 \leq 1}} \|\text{vec}(N)\|_\infty \quad \text{e} \quad \vartheta(M) \stackrel{\text{def}}{=} \max_{\substack{N \in \Omega(M), \\ \|\text{vec}(N)\|_\infty \leq 1}} \|N\|_2, \quad (2.36)$$

em que  $T(M)$  é a variedade de todas as matrizes com posto menor que ou igual a  $\text{rank}(M)$ , e  $\Omega(M)$  é o espaço tangente em  $M$  com relação à variedade de todas as matrizes com o número de entradas não nulas menor do que ou igual a  $\text{nnz}(M)$ .

O principal resultado de recuperabilidade é o Teorema 1 a seguir. A condição suficiente principal de todo o trabalho é a de que o produto  $\vartheta(S^*)\xi(L^*)$  seja menor do que  $\frac{1}{6}$ . Observe, como mencionamos anteriormente, que os cálculos são feitos com as componentes da solução  $(S^*, L^*)$ . Note, também, que os autores fornecem um valor específico para o parâmetro  $\gamma$  que, como antes, depende da solução.

**Teorema 1** ((Teo. 2 de [25]<sup>6</sup>) Unicidade de Solução.). Dadas  $A = S^* + L^*$  com

$$\vartheta(S^*)\xi(L^*) < \frac{1}{6}, \quad (2.37)$$

o único minimizador  $(\hat{S}, \hat{L})$  de (2.5) é  $(S^*, L^*)$ , para o seguinte intervalo de  $\gamma$ :

$$\gamma \in \left( \frac{\xi(L^*)}{1 - 4\vartheta(S^*)\xi(L^*)}, \frac{1 - 3\vartheta(S^*)\xi(L^*)}{\vartheta(S^*)} \right). \quad (2.38)$$

Em particular,

$$\gamma = \sqrt{\frac{3\xi(L^*)}{2\vartheta(S^*)}} \quad (2.39)$$

sempre pertence ao intervalo acima e, portanto, garante a recuperação exata de  $(S^*, L^*)$ .

O que os autores demonstram é que, se (2.37) vale, então as condições de otimalidade que garantem a unicidade da solução também são satisfeitas.

Antes de enunciar as condições de otimalidade de Chandrasekaran *et al.* [25] para o Problema (2.5), precisamos definir os espaços associados às soluções  $S^*$  e  $L^*$  que aparecem em seu enunciado e, em particular, em (2.36). Para cada matriz, há dois espaços complementares associados e as duas respectivas matrizes de projeção. A fim de evitar notação carregada, sempre que possível omitiremos os pontos  $S^*$  e  $L^*$  em que cada espaço é definido.

Seja  $\Omega$  o espaço de todas as matrizes que têm o mesmo suporte que  $S^*$ :

$$\Omega = \{M \in \mathbb{R}^{m \times n} \mid \text{supp}(M) \subseteq \text{supp}(S^*)\}. \quad (2.40)$$

A projeção de uma matriz no espaço  $\Omega$  é denotada por  $P_\Omega$  e sua ação anula as entradas que não estão em  $\text{supp}(S^*)$ . Seja  $\Omega^c$  o complemento ortogonal a  $\Omega$ , ou seja, o espaço das matrizes que têm suporte em  $\text{supp}(S^*)^c$ . A projeção em  $\Omega^c$  é denotada por  $P_{\Omega^c}$ .

Considere a SVD reduzida  $L^* = U\Sigma V^\top$ , em que  $k$  é o posto de  $L^*$ , com  $U \in \mathbb{R}^{m, \times k}$ ,  $V^\top \in \mathbb{R}^{n \times k}$ . Seja, também, o espaço tangente à variedade das matrizes de posto menor ou igual a  $\text{rank}(L^*)$  em  $L^*$ :

$$T = \{UX^\top + YV^\top \mid X \in \mathbb{R}^{n \times k}, Y \in \mathbb{R}^{m \times k}\}. \quad (2.41)$$

<sup>6</sup> Traduzido e adaptado.

$T$  é o espaço gerado por todas as matrizes com espaço linha contido no espaço linha de  $L^*$  ou com o espaço coluna contido no espaço coluna de  $L^*$ . A projeção de uma matriz no espaço  $T$  é dado por  $P_T$ :

$$P_T(M) = P_U M + M P_V - P_U M P_V, \quad (2.42)$$

em que  $P_U = U U^\top$  e  $P_V = V V^\top$  são as projeções nos espaços gerados por  $U$  e  $V$ , respectivamente. Ainda, seja  $T^\perp$  o espaço ortogonal a  $T$ , isto é, o espaço gerado por todas as matrizes com espaço coluna ortogonal ao espaço coluna de  $L^*$  e espaço linha ortogonal ao espaço linha de  $L^*$ . Assim, a projeção nesse espaço é dada por  $P_{T^\perp}$ :

$$P_{T^\perp}(M) = (I_n - P_U)M(I_n - P_V). \quad (2.43)$$

Com as devidas definições estabelecidas, Chandrasekaran *et al.* enunciam as condições de otimalidade para o Problema (2.5), e fornecem o esquema da Figura 1 para ajudar no entendimento do que cada condição representa.

As condições de otimalidade na Proposição 1 aplicadas ao Problema (2.5) nos permitem afirmar que  $(S^*, L^*)$  é solução ótima de (2.5) se e somente se existe uma variável dual  $Z \in \mathbb{R}^{m \times n}$  tal que

$$Z \in \gamma \partial \|S^*\|_1 \quad \text{e} \quad Z \in \partial \|L^*\|_*. \quad (2.44)$$

**Proposição 2** (Proposição 2 de [25]<sup>7</sup>). Suponha que  $A = S^* + L^*$ . Então  $(S, L) = (S^*, L^*)$  é a única solução do Problema (2.5) se as seguintes condições são satisfeitas:

- i.  $\Omega(S^*) \cap T(L^*) = \{0\}$
- ii. Existe uma variável dual  $Z \in \mathbb{R}^{m \times n}$  tal que
  - a)  $P_{T(L^*)}(Z) = UV^\top$ ;
  - b)  $P_{\Omega(S^*)}(Z) = \gamma \text{sign}(S^*)$ ;
  - c)  $\|P_{T(L^*)^\perp}(Z)\| \leq 1$ ;
  - d)  $\|P_{\Omega(S^*)^c}(Z)\|_\infty < \gamma$ .

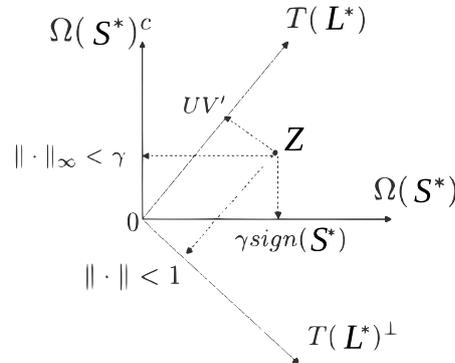
A esta altura, caracterizado o subdiferencial das normas da função objetivo, podemos invocar a Proposição 1, em particular a expressão dada em (2.28), para aplicar as condições de otimalidade para o Problema (2.5).

**Proposição 3 (Condições Suficientes de Otimalidade para (2.5)).** Um ponto  $(\hat{S}, \hat{L})$  é um ótimo do Problema (2.5) se existe um certificado dual  $Z \in \mathbb{R}^{m \times n}$  tal que

$$Z \in \gamma \partial \|\hat{S}\|_1 \quad \text{e} \quad Z \in \partial \|\hat{L}\|_*. \quad (2.45)$$

<sup>7</sup> Traduzida e adaptada.

Figura 1 – Figura adaptada de [25, Fig. 4.1]. Representação geométrica das condições suficientes de otimalidade para o Problema (2.5): existência de um ponto dual  $Z$ . As setas tracejadas indicam projeções ortogonais. Cada projeção deve satisfazer uma condição, descrita próxima a cada seta tracejada, em concordância com a Proposição 2.



Para  $Z \in \gamma \partial \|\hat{S}\|_2$ , precisamos ter

$$P_{\Omega}(Z) = \gamma \operatorname{sign}(\hat{S}), \quad \text{e} \quad \|P_{\Omega^c}(Z)\|_{\infty} < \gamma. \quad (2.46)$$

Para  $Z \in \partial \|\hat{L}\|_*$ , precisamos ter ([99])

$$P_T(Z) = UV^T, \quad \text{e} \quad \|P_{T^c}(Z)\| \leq 1. \quad (2.47)$$

Como podemos ver, a Proposição 3 corresponde às condições (ii) da Proposição 2. A condição (i) é usada para demonstrar a *unicidade* da solução, em sentido que discutimos a seguir.

### 2.5.1 Algumas considerações

O Teorema 1 fornece condições determinísticas suficientes para que se garanta a unicidade da solução do Problema (2.5) sem a necessidade de hipóteses sobre o padrão de esparsidade de  $S^*$  nem sobre o posto de  $L^*$ . Em termos didáticos, a desigualdade (2.37) pode ser interpretada com o significado de que os espaços tangentes  $\Omega(S^*)$  e  $T(L^*)$  são *suficientemente* transversais.

Entretanto, reiteramos que as medidas  $\vartheta(L^*)$  e  $\xi(S^*)$  dependem do conhecimento prévio da solução e não têm, portanto, um caráter prático. Um agravante é que o próprio intervalo de  $\gamma$  depende dessas medidas – e calculá-las pode significar resolver outros problemas de otimização, possivelmente de difícil resolução.

Além disso, os autores esclarecem que, no caso geral de uma matriz  $A = S^* + L^*$ , há de fato decomposições alternativas de uma mesma  $A$  com diferentes componentes “esparsa” e de “posto baixo”. Por exemplo, em um caso geral, é

sempre possível subtrair de  $S^*$  uma entrada não nula e adicioná-la a  $L^*$  mantendo a factibilidade, aumentando o posto da nova  $L^*$  em no máximo 1 e diminuindo estritamente a esparsidade de  $S^*$ .

Suponha que após a mudança temos  $S_0$  e  $L_0$ . Nesse caso, se pudermos recalcular os valores de  $\vartheta$  e  $\xi$ , veremos que  $\vartheta(S_0)$  diminuiu e  $\xi(L_0)$  possivelmente aumentou bastante – a ponto de compensar a leve diminuição de  $\vartheta$  –, já que cada valor pode ser interpretado como uma medida de quão “difusos” são os padrões de esparsidade e posto dessas matrizes, respectivamente. Os autores argumentam que, embora  $(S_0, L_0)$  seja uma decomposição do tipo esparsa mais posto baixo, não é a decomposição mais difusa, na medida em que  $L_0$  teve comprometida sua “difusividade”. Um raciocínio análogo valeria se passássemos uma unidade de posto de  $L^*$  para  $S^*$  (por meio de uma parcela de posto 1 da SVD, por exemplo).

Ainda segundo os autores, o resultado de recuperabilidade deve ser interpretado da seguinte forma: Dada uma matriz  $A = S^* + L^*$  formada por uma matriz esparsa  $S^*$  com suporte difuso e uma matriz  $L^*$  de posto baixo com espaços linha e coluna difusos, o Problema (2.5) irá recuperar essa decomposição *difusa*, e que isto é exatamente o que alguns aplicações esperam do problema (veja as observações finais em [25, p. 582]).

Uma outra abordagem

Uma outra abordagem semelhante para o tema da existência e unicidade de solução pode ser encontrada em Candès *et al.* [21]. Partindo de outro contexto de aplicação e de hipóteses levemente diferentes, esses autores conseguem resultados que se diferenciam dos que enunciamos anteriormente em pelo menos dois aspectos importantes.

O primeiro deles é que não são resultados determinísticos: a recuperação exata é garantida com alta probabilidade e, na demonstração, há a exigência de que a localização das entradas não nulas de  $S^*$  sejam sorteadas e uniformemente distribuídas pela matriz. O segundo, com importantes consequências práticas, é que a demonstração também é válida pra um valor universal da constante  $\gamma$ , a saber: o valor do parâmetro no caso  $m \times n$  é.<sup>8</sup>

$$\gamma = \frac{1}{\sqrt{\max\{m, n\}}}. \quad (2.48)$$

Os autores argumentam que esse valor não está associado de antemão ao desempenho dos métodos e, para problemas práticos, ajustes empíricos podem proporcionar melhores resultados.

<sup>8</sup> Para  $t \in (0, 1)$  e  $\gamma \in (0, \infty)$  temos que  $\gamma = \frac{t}{1-t} \iff t = \frac{\lambda}{\lambda+1}$ .

As condições de recuperação exata estabelecidas nesse caso envolvem a hipótese de que a componente  $L^*$  satisfaz uma condição batizada de *condição de incoerência*, a qual também estabelece que a matriz de posto baixo não pode ser esparsa – ou, mais precisamente, que seus vetores singulares estejam *razoavelmente* espalhados.

Neste trabalho, como estamos interessados principalmente na aplicação do problema em instâncias com filmagens reais, as condições suficientes de recuperabilidade não influenciarão diretamente os testes computacionais. Como veremos, a versão equivalente da expressão (2.48) para o parâmetro  $t$  será adotada e fixada em todos os experimentos.

## 2.6 Caracterização via Programação Semidefinida

Como mencionamos na introdução, o Problema (2.5) pode ser reformulado como um problema de programação semidefinida (SDP) equivalente e, assim, resolvido por algum método de pontos interiores, como fizeram os autores em [25]. Apesar da dificuldade de lidar com problemas de grande porte, a reformulação pode ser útil, por exemplo, para exploração de problemas-testes artificiais. Há alguns *softwares* especializados que são disponibilizados gratuitamente, como o YALMIP ([64]) e o CVX ([45]), para MATLAB ou, no caso do YALMIP, também para Octave.

Começamos definindo a norma dual. O produto interno matricial corresponde ao produto interno de Frobenius no espaço das matrizes reais, o qual pode ser visto como o produto interno usual das matrizes reorganizadas como vetores:

$$\langle M, N \rangle = \text{tr}(M^T N) = \text{vec}(M)^T \text{vec}(N). \quad (2.49)$$

**Definição 4** (Norma Dual). Seja  $\|\cdot\|$  uma norma em  $\mathbb{R}^{m \times n}$ . A norma dual  $\|\cdot\|_d$  associada a  $\|\cdot\|$  é definida para todo  $Y \in \mathbb{R}^{m \times n}$  em termos do produto interno de Frobenius como

$$\|Y\|_d \stackrel{\text{def}}{=} \sup_{\|X\| \leq 1} \langle Y, X \rangle = \sup_{\|X\| \leq 1} \text{tr}(X^T Y). \quad (2.50)$$

Conforme consta em [51, Seção 5.6], são identidades alternativas às de (2.50):

$$\|Y\|_d = \sup_{\|X\| \leq 1} |\langle Y, X \rangle| = \sup_{\|X\|=1} |\langle Y, X \rangle| = \sup_{\|X\|=1} \langle Y, X \rangle = \sup_{X \neq 0} \frac{\langle Y, X \rangle}{\|X\|} = \max_{\|X\| \leq 1} \langle Y, X \rangle, \quad (2.51)$$

dentre outras variações semelhantes a essas.

Não é difícil provar que  $\|\cdot\|_d$  também é uma norma em  $\mathbb{R}^{m \times n}$ , a partir da própria definição de norma. Assim, por exemplo, podemos mostrar que a norma dual

da norma nuclear em  $\mathbb{R}^{m \times n}$  é a norma 2 matricial (também conhecida como norma espectral). Veja [17, Seção A.1.6], [51, pp. 360, 464].

**Proposição 4** (Desigualdade de Cauchy-Schwarz Generalizada). Da definição de norma dual  $\|\cdot\|_d$  de uma norma  $\|\cdot\|$  segue que

$$|\langle Y, X \rangle| \leq \|Y\|_d \|X\|. \quad (2.52)$$

O fato de que a dual de uma norma dual é a norma original é um dos resultados mais importantes da dualidade de normas. Esse resultado permite representar uma norma  $\|\cdot\|$  através de sua dual  $\|\cdot\|_d$  da seguinte maneira:

$$\|X\| = \max_{\|Y\|_d=1} \langle X, Y \rangle, \quad (2.53)$$

o que é uma exemplo de *quase-linearização* (cf. [51, Equação 5.5.10]). No caso da norma 2 (espectral) em  $\mathbb{R}^{m \times n}$ , resulta que sua norma dual é a norma nuclear ([17, Seção A.1.6], [51, 360, 464]).

**Proposição 5** (Dual da norma 2 matricial). A dual da norma 2 matricial é a norma nuclear.

*Demonstração.* A demonstração será feita em duas partes. Na primeira, mostramos que a norma dual de uma matriz  $X$  qualquer nunca é menor do que a norma nuclear dessa mesma matriz. Por outro lado, na segunda parte, mostramos que a norma dual nunca é maior do que a norma nuclear para uma mesma matriz. Assim, conclui-se que ambas as normas têm o mesmo valor e, conseqüentemente, são iguais.

Parte 1. Seja  $X$  uma matriz qualquer e  $U\Sigma V^\top$  sua decomposição em valores singulares (SVD). Escolhendo  $Y = UV^\top$  na Definição 4, podemos ver que a norma dual é maior ou igual à norma nuclear, isto é:

$$X = U\Sigma V^\top, Y = UV^\top \implies \boxed{\|X\|_d \geq \|X\|_*}.$$

Parte 2. A ideia é encontrar um limitante superior para o valor ótimo do problema de otimização expresso em (2.50) através de seu problema dual em programação semidefinida e um ponto dual-factível. Para isso, iniciamos por reescrever a norma 2 matricial através de desigualdades generalizadas:

$$\|Z\|_2 \leq t \iff t^2 I_n - ZZ^\top \geq 0. \quad (2.54)$$

Como  $S = tI_n - (\frac{1}{t})ZZ^\top$  é o complemento de Schur da matriz  $\begin{bmatrix} tI_n & Z \\ Z^\top & tI_n \end{bmatrix}$  com relação ao bloco diagonal inferior, e  $S \geq 0$  por (2.54) (além do caso trivial  $tI_n > 0$ ), temos a garantia ([17, Apêndice A.5.5]) de que

$$t^2 I_n - ZZ^\top \geq 0 \iff \begin{bmatrix} tI_n & Z \\ Z^\top & tI_n \end{bmatrix} \geq 0. \quad (2.55)$$

Portanto,

$$\|Z\|_2 = \inf \left\{ t; \begin{bmatrix} tI_n & Z \\ Z^\top & tI_n \end{bmatrix} \geq 0 \right\}, \quad (2.56)$$

e, assim, obtemos a equivalência

$$\begin{aligned} \max \quad & \text{tr}(X^\top Y) \\ \text{s. a:} \quad & \|Y\| \leq 1 \end{aligned} \iff \begin{aligned} \max \quad & \text{tr}(X^\top Y) \\ \text{s. a:} \quad & \begin{bmatrix} I_n & Y \\ Y^\top & I_n \end{bmatrix} \geq 0. \end{aligned} \quad (2.57)$$

Da programação semidefinida, construímos, por fim, o problema dual (Lagrangiano) àquele que aparece em (2.50):

$$\begin{aligned} \text{minimizar} \quad & \frac{1}{2} \text{tr}(W_1 + W_2) \\ \text{sujeito a} \quad & \begin{bmatrix} W_1 & X \\ X^\top & W_2 \end{bmatrix} \geq 0. \end{aligned} \quad (2.58)$$

Se  $X = U\Sigma V^\top$ ,  $\bar{W}_1 = U\Sigma U^\top$  e  $\bar{W}_2 = V\Sigma V^\top$ , então  $(\bar{W}_1, \bar{W}_2, X)$  é dual factível. Portanto,  $\frac{1}{2} \text{tr}(\bar{W}_1 + \bar{W}_2) \equiv \|X\|_\star$  é um limitante superior para o valor ótimo, isto é:

$$\boxed{\|X\|_\star \geq \|X\|_d}. \quad (2.59)$$

Como não há *gap* de dualidade, a consequência é que  $\|X\|_\star$  pode ser calculada tanto pelo problema primal quanto pelo problema dual. ■

A Proposição 5 nos leva à igualdade

$$\begin{aligned} \|X\|_\star = \min \quad & \frac{1}{2} \text{tr}(W_1 + W_2) \\ \text{s. a:} \quad & \begin{bmatrix} W_1 & X \\ X^\top & W_2 \end{bmatrix} \geq 0 \end{aligned} \quad (2.60)$$

e, consequentemente, ao problema equivalente

$$\min_{S, L, W_1, W_2} \gamma \|S\|_1 + \frac{\text{tr}(W_1 + W_2)}{2} \quad (2.61)$$

$$\text{s. a:} \quad \begin{bmatrix} W_1 & L \\ L^\top & W_2 \end{bmatrix} \geq 0 \quad (2.62)$$

$$S + L = A. \quad (2.63)$$

Observe que, para calcular a norma nuclear de uma matriz dessa forma, precisamos resolver um problema de otimização com desigualdades generalizadas em  $(m+n)^2$  variáveis.

## 2.7 Aproximações Suaves para as Normas via Dualidade

Neste momento, passamos a considerar a suavização de um dos termos da função objetivo do Problema (2.5), com o intuito de ampliar as possibilidades de métodos computacionais e adotar estratégias que têm se mostrado exitosas na resolução de problemas com um termo suave e outro convexo. Com esse propósito, invocamos algumas fórmulas para o cálculo das derivadas da norma 1 e nuclear. Embora, na prática, a suavização que adotamos nos experimentos seja a da norma 1, exibimos os resultados para a norma nuclear por completude e para possíveis trabalhos futuros.

O conceito de função proximal [74, p. 130], junto com a Definição 2, nos permitirão reescrever a função objetivo de maneira apropriada para a suavização.

**Definição 5** (Função-prox). Uma função  $d$  é chamada de função-prox se  $d$  é uma função contínua fortemente convexa em seu domínio com parâmetro de convexidade forte  $\sigma > 0$ .

No início, adotamos algumas estratégias para suavizar a função objetivo convexa do Problema (2.5). No entanto, continuamos a procurar por uma maneira mais geral de utilizar aproximações suaves e acabamos encontrando o trabalho [74], no qual esse assunto é abordado, e o [7, Exercício 5.4.5], que se trata de um caso particular da mesma abordagem.

Em [74], Nesterov essencialmente estabelece que se uma função objetivo não suave  $f$  é a conjugada de alguma função convexa fechada  $g$  com domínio limitado (isto é, se  $f$  admite uma representação conjugada), então uma aproximação suave de  $f$  pode ser construída através de uma função proximal auxiliar  $d$ . Em termos matemáticos, se  $f$  pode ser escrita como

$$f(x) = g^*(x) \stackrel{\text{def}}{=} \sup_{u \in \text{dom}(g)} \{ \langle u, x \rangle - g(u) \}, \quad (2.64)$$

então

$$f_\mu(x) = (g + \mu d)^*(x) \quad (2.65)$$

$$\stackrel{\text{def}}{=} \sup_{u \in \text{dom}(g)} \{ \langle u, x \rangle - (g(x) + \mu d(x)) \} \quad (2.66)$$

é uma aproximação suave de  $f$  com parâmetro  $\mu$ , em que  $(g + \mu d)^*$  é a conjugada da função  $g + \mu d$ . Além disso, é possível mostrar que  $\nabla f_\mu(x)$  satisfaz uma condição de Lipschitz com constante no máximo  $\mu^{-1}$ . Como as únicas exigências sobre a função proximal  $d$  são a continuidade e a convexidade forte, é fácil ver que há infinitas possibilidades para  $d$ .

Em se tratando de uma norma qualquer, podemos usar o conceito de norma dual para representá-la, como em (2.64). Como nosso interesse está principalmente nas normas matriciais, é para esse tipo de normas que desenvolveremos um pouco mais a dualidade.

Da Definição 4 de norma dual vemos que é possível expressar uma norma matricial  $\|\cdot\|$  em termos de sua norma dual da seguinte maneira:

$$\|x\| = \sup_{\|u\|_d \leq 1} \langle u, x \rangle.$$

Isso nos permite identificar uma norma com o formato (2.64) fazendo  $B = \{u \mid \|u\|_d \leq 1\}$  e  $g(u) = 0$  em  $B$ , além do estabelecimento de alguma função proximal  $d$ , o que nos leva à aproximação suave

$$\|x\| \approx f_\mu(x) = \sup_{\|u\|_d \leq 1} \{\langle u, x \rangle - \mu d(x)\}. \quad (2.67)$$

### 2.7.1 Aproximação Suave de Primeira Ordem: a função de Huber

Consideremos as seguintes funções-prox para as normas 1 e nuclear, respectivamente:

$$d(u) = \frac{1}{2} \|u\|_2^2 \quad \text{e} \quad d(U) = \frac{1}{2} \|U\|_F^2.$$

É possível mostrar que, com essas escolhas, e usando a Equação (2.67), chegamos respectivamente às aproximações (cf. [63, 1237])

$$\|x\|_1 \approx \sup_{\|u\|_\infty \leq 1} \left\{ u^\top x - \frac{\mu}{2} \|u\|_2^2 \right\} = \sum_{i=1}^n h(x_i; \mu) \quad (2.68)$$

$$\|X\|_* \approx \sup_{\|U\|_2 \leq 1} \left\{ \text{tr}(U^\top X) - \frac{\mu}{2} \|U\|_F^2 \right\} = \sum_{i=1}^{\text{rank}(X)} h(\sigma_i(X); \mu), \quad (2.69)$$

em que  $h$  é a função de Huber univariada com parâmetro  $\mu$ , definida por

$$h(x; \mu) \stackrel{\text{def}}{=} \begin{cases} \frac{x^2}{2\mu}, & \text{se } |x| \leq \mu \\ |x| - \frac{\mu}{2}, & \text{se } |x| \geq \mu \end{cases} \quad (2.70)$$

e ilustrada na Figura 2 junto com outras funções relevantes.

Por conveniência de notação, definimos a função  $h_\mu : \mathbb{R}^n \rightarrow \mathbb{R}$  como a soma das aplicações da função de Huber com parâmetro  $\mu$  em cada componente do vetor  $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$ , ou seja,

$$h_\mu(x) \stackrel{\text{def}}{=} \sum_{i=1}^n h(x_i; \mu). \quad (2.71)$$

Com isso, podemos reescrever as Equações (2.68) e (2.69) para  $x \in \mathbb{R}^n$  e  $X \in \mathbb{R}^{m \times n}$  como

$$\|x\|_1 \approx h_\mu(x) \quad \text{e} \quad \|X\|_* \approx h_\mu(\sigma(X)), \quad (2.72)$$

em que  $(\sigma_1(X), \sigma_2(X), \dots, \sigma_r(X), 0, \dots, 0) \in \mathbb{R}^{\min(m,n)}$ , sendo  $r = \text{rank}(X)$ .

É possível mostrar que a função de Huber, como aproximação (suave) para norma 1, satisfaz a desigualdade

$$h_\mu(x) \leq \|x\|_1 \leq h_\mu(x) + \frac{\mu n}{2}.$$

Com isso, fica claro que quanto menor o parâmetro  $\mu$ , melhor é a aproximação.

Além disso, a derivada da função de Huber univariada pode ser facilmente calculada através da expressão

$$\frac{dh(x_i; \mu)}{dx_i} = \begin{cases} \frac{x_i}{\mu}, & \text{se } |x_i| \leq \mu \\ \text{sign}(x_i), & \text{se } |x_i| \geq \mu \end{cases}. \quad (2.73)$$

No contexto dos problemas de regressão em que se deseja minimizar a norma do resíduo de um sistema linear e, em particular, no problema de quadrados mínimos lineares, a função de Huber é adotada como penalização dos resíduos por apresentar características intermediárias entre a normas 1 e 2, sendo considerada uma aproximação robusta para ambas ([17, Seção 6.1]).

Como podemos ver por sua definição, trata-se de uma função quadrática próxima ao ponto de avaliação acoplada a uma função valor absoluto em regiões mais distantes, sendo regida pelo parâmetro  $\mu$ , que tipicamente é menor do que 0.5. Isso faz com que a função de Huber lide melhor com pontos destoantes (*outliers*) do que a norma 2, já que, comparativamente, atribui menos peso a grandes valores na minimização de resíduos, pois  $h(x; \mu) \ll x^2$  para valores grandes de  $x$ . Em contrapartida, para valores pequenos de  $x$ , a função de Huber coloca menos ênfase do que a norma 1, pois  $h(x; \mu) \ll |x|$  neste caso. Portanto, ao adotar a função de Huber é natural que se espere ter resquícios de resíduos, e não soluções rigorosamente esparsas, como ocorre com a norma 1.

Para uma discussão mais ampla, veja, em particular, a Seção 6.1.2 de [17], com a ressalva de que Boyd e Vandenberghe definem  $\phi_{\text{hub}}$  como função de Huber e, na nossa notação,  $\phi_{\text{hub}}(x) = 2\mu h(x; \mu)$ . Sobre a robustez das aproximações feitas com diferentes suavizações, o próprio estatístico suíço Peter J. Huber analisa suas propriedades em [54, 55].

### 2.7.1.1 Aproximação de Classe $C^1$

Usando as aproximações (2.68), (2.69) e (2.72) para as normas 1 e nuclear em termos da função de Huber, substituindo  $S$  por  $A - L$  e, simplificando a identificação

da matriz de posto baixo através de  $X = (x_{ij})$ , obtemos o seguinte modelo suave de aproximação para a função objetivo  $\Psi$  de (2.7):

$$\Psi_\mu(X) = t \sum_{i=1}^m \sum_{j=1}^n h(x_{ij} - a_{ij}; \mu) + (1-t) \sum_{i=1}^{\text{rank}(X)} h(\sigma_i(X); \mu) \quad (2.74)$$

$$= t h_\mu(X - A) + (1-t) h_\mu(\sigma(X)). \quad (2.75)$$

Empilhando as entradas da matriz  $X$  por colunas, é possível interpretar as variáveis como um vetor  $x$  em  $\mathbb{R}^{mn}$ . Assim, o gradiente de  $\Psi_\mu$  em  $X$  pode ser expresso componente a componente por

$$\begin{aligned} [\nabla \Psi_\mu(X)]_k &\stackrel{\text{def}}{=} \frac{\partial \Psi_\mu(x)}{\partial x_k} \\ &= t \frac{dh}{dx_k}(x_k - a_k; \mu) + (1-t) \sum_{i=1}^{\text{rank}(X)} \frac{dh}{dx_k}(\sigma_i(X); \mu) [\text{vec}(u_i v_i^\top)]_k, \end{aligned} \quad (2.76)$$

$$k = 1, 2, \dots, mn,$$

em que  $u_i$  e  $v_i$  são as colunas correspondentes da matriz  $U$  e  $V$  da decomposição de  $X$  em valores singulares.

Por sua vez, a derivada da função composta  $(h_\mu \circ \sigma)(X)$  com relação às entradas de  $X$ , obtida de [61], será exibida na Seção 2.7.3.

## 2.7.2 Aproximação de Segunda Ordem: a função pseudo-Huber

Vimos que a função de Huber é diferenciável apenas até a primeira ordem e isso será suficiente para os experimentos deste trabalho. No entanto, caso se deseje utilizar métodos numéricos de segunda ordem, precisamos de uma função de aproximação que possua ao menos as segundas derivadas. Apenas para direcionar o leitor com interesse nessa possibilidade, consideramos a versão alternativa à função de Huber, conhecida como função pseudo-Huber com parâmetro  $\mu$  ([39, Section 1.1]):

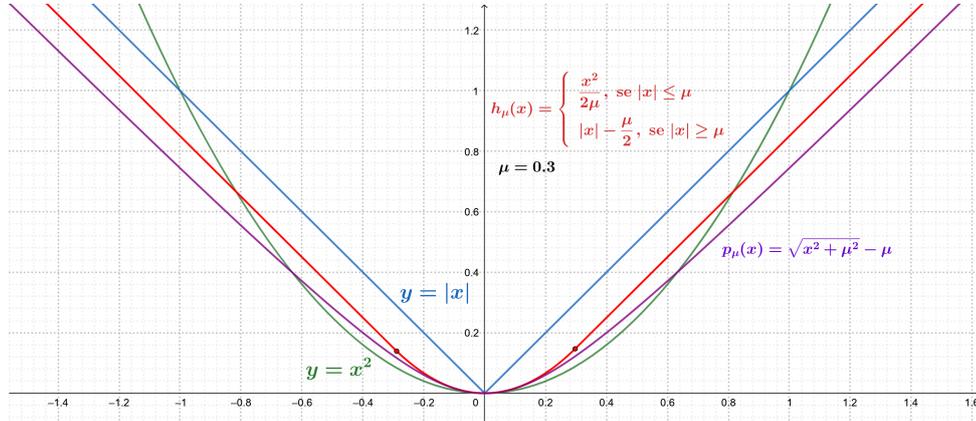
$$p(x; \mu) = \sqrt{\mu^2 + x^2} - \mu, \quad (2.77)$$

para algum  $\mu > 0$  fixado.

Diferente da função de Huber, a pseudo-Huber tem derivadas parciais de todas as ordens. Derivando a expressão acima, obtemos as derivadas de ordem 1 e 2 como

$$\frac{dp(x; \mu)}{dx} = \frac{x}{\sqrt{x^2 + \mu^2}} \quad \text{and} \quad \frac{d^2p(x; \mu)}{dx^2} = \frac{\mu}{(x^2 + \mu^2)^{3/2}}. \quad (2.78)$$

Figura 2 – Aproximações suaves para o valor absoluto (azul) via funções de Huber (vermelho) e pseudo-Huber (roxo) adotando  $\mu = 0.3$ , além da função quadrática pura (verde) para comparação. A função pseudo-Huber é, ponto a ponto, menor do que ou igual à função de Huber, a qual por sua vez subestima a função módulo, também ponto a ponto. Imagem feita com o Geogebra [50].



Agora, somos capazes de escrever as expressões para a função  $\hat{\Psi}_\mu$  e seu gradiente, analogamente ao que fizemos para a função de Huber.

$$\hat{\Psi}_\mu(X) = t \sum_{i=1}^m \sum_{j=1}^n p(x_{ij} - a_{ij}; \mu) + (1 - t) \sum_{i=1}^{\text{rank}(X)} p(\sigma_i(X); \mu),$$

$$[\nabla \hat{\Psi}_\mu(X)]_k = t \frac{dp}{dx_k}(x_k - a_k; \mu) + (1 - t) \sum_{i=1}^{\text{rank}(X)} \frac{dp}{dx_k}(\sigma_i(X); \mu) [\text{vec}(u_i v_i^\top)]_k, \quad (2.79)$$

$$k = 1, 2, \dots, mn,$$

em que  $u_i$  e  $v_i$  são as colunas correspondentes das matrizes  $U$  e  $V$  da SVD de  $X$ . Como antes, a derivada da função composta  $(p \circ \sigma)(X)$  com relação às entradas de  $X$  foi obtida em [61] e será exibida na Seção 2.7.3.

Para ilustrar o comportamento dessas funções, apresentamos na Figura 2 uma comparação visual entre aproximações via Huber, pseudo-Huber, função módulo e função quadrática.

### 2.7.3 Derivadas das Aproximações Suaves

A fim de determinar expressões para as derivadas de uma função  $\phi$  que pode ser expressa como a composição  $(g \circ \sigma)$  de uma função absolutamente simétrica  $g : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$  com o vetor  $\sigma(X)$  de valores singulares de  $X \in \mathbb{R}^{n \times n}$  ordenados em ordem decrescente, recorreremos à teoria de funções de valores singulares, também conhecidas como funções ortogonalmente invariantes (veja [61, Proposição 5.1]).

De acordo com a Proposição 6.2 em [61, p. 230], dada uma função  $g$  convexa e absolutamente simétrica, a composição  $(g \circ \sigma)(X)$  é diferenciável em  $X$  se e somente

se  $g$  é diferenciável em  $\sigma(X)$ . O gradiente da função composta é então dado por

$$\nabla(g \circ \sigma)(X) = U (\text{Diag } \nabla g(\sigma(X))) V^\top, \quad (2.80)$$

para quaisquer matrizes ortogonais com entradas reais  $U$  e  $V$ , com  $X = U (\text{Diag } \sigma(X)) V^\top$ .

No nosso caso, sendo  $h : \mathbb{R} \rightarrow \mathbb{R}$  a função de aproximação do tipo Huber ou pseudo-Huber, temos que

$$g(x) = \sum_i h(x_i; \mu) \quad \text{e} \quad \phi(X) = (g \circ \sigma)(X) \quad (2.81)$$

satisfazem todas as condições anteriores, de forma que podemos calcular seu gradiente por meio da expressão

$$\nabla \phi(X) = U (\nabla g(\sigma(X))) V^\top, \quad (2.82)$$

a qual é válida para quaisquer matrizes ortogonais com entradas reais  $U$  e  $V$ , com  $X = U (\text{Diag } \sigma(X)) V^\top$ .

#### 2.7.4 Alguns comentários

Vimos que, em teoria, a diminuição do valor do parâmetro  $\mu$  faz com que a aproximação da função objetivo seja cada vez melhor. No entanto, como destacam os autores em [39, Section 1.1], existe uma relação de compromisso na escolha desse parâmetro. Na prática, valores muito pequenos podem conduzir a instabilidades numéricas fatais.

Por outro lado, expressões para as derivadas da composição da função Huber ou pseudo-Huber com o vetor  $\sigma$  de valores singulares não são trivialmente dedutíveis. Em certo sentido, a regra da cadeia parece não desempenhar papel algum na fórmula final das derivadas, como vimos na Seção 2.7.3.

Em nossos experimentos computacionais, aplicamos as técnicas desta seção para suavizar a norma 1 por meio da função de Huber, deixando a norma nuclear intacta. No entanto, as expressões para a suavização da norma nuclear também podem ser adotadas em trabalhos futuros – quer o objetivo seja suavizar ambos os termos da função, quer seja suavizar apenas o termo da norma nuclear.

## 2.8 Melhores Aproximações para uma Matriz via SVD

O problema de que tratamos envolve a decomposição de uma dada matriz como a soma de duas outras com características especiais. Dependendo do contexto de aplicação do problema, é possível que uma das parcelas possa ser interpretada como uma matriz de erros, grandes ou não. Nesse caso, um questionamento que

surge naturalmente é se podemos extrair da própria matriz  $A$  a *essência* de uma das parcelas incógnitas, de modo a se aproximar da solução do problema apenas com *alguma* manipulação dos dados de entrada.

Embora não haja garantias de sucesso no caso geral, esse questionamento nos conduz às aproximações de uma matriz a partir de seus valores e vetores singulares, clássica da álgebra linear computacional. Nesse sentido, vale destacar que a SVD pode ser interpretada como uma decomposição matricial em termos de uma soma de matrizes de posto unitário. O teorema que apresentamos abaixo, nomeado em homenagem a Eckart, Young e Mirsky, nos auxiliará a implementar essas ideias computacionalmente no Capítulo 4, em particular na Seção 4.5, e na discussão em aberto sobre a escolha do ponto inicial para o problema SLRMD, no Capítulo 8.

**Teorema 2** (Eckart-Young-Mirsky). Considere  $\|\cdot\|$  uma norma unitariamente invariante<sup>9</sup>. Seja  $A \in \mathbb{R}^{m \times n}$  uma matriz de posto  $r$  com valores singulares  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$  e vetores singulares  $u_1, u_2, \dots, u_r$ , à esquerda, e  $v_1, v_2, \dots, v_r$ , à direita, de forma que

$$A = \sum_{i=1}^r \sigma_i u_i v_i^\top. \quad (2.83)$$

Se

$$k < r \quad \text{e} \quad A_k = \sum_{i=1}^k \sigma_i u_i v_i^\top, \quad (2.84)$$

então

$$\min_{\substack{\text{rank}(X)=k \\ X \in \mathbb{R}^{m \times n}}} \|A - X\| = \|A - A_k\|. \quad (2.85)$$

De acordo com Strang [92, Seção I.9], o Teorema 2 tem suas origens no trabalho de [89], na ocasião em que foi apresentado para operadores  $A$  em um espaço de funções, diretamente extensível a matrizes em espaços vetoriais. O enunciado para a norma de Frobenius, desta vez com uma nova demonstração, foi publicado por Eckart e Young [31]. A versão que apresentamos aqui é válida para qualquer norma unitariamente invariante – como, por exemplo, qualquer norma que dependa apenas dos valores singulares e, em particular, a norma nuclear – e foi adaptada na generalização apresentada e demonstrada por Mirsky em [69, Teorema 3]. Outra publicação contemporânea a essa com resultados semelhantes é o artigo de Ky Fan e Hoffman [34].

Vemos, assim, que a matriz  $A_k$  do Teorema 2 é de grande relevância quando se fala em aproximações para  $A$ . A fim de facilitar a menção a essa matriz, apresentamos a definição a seguir.

<sup>9</sup> Uma norma  $\|\cdot\|$  em  $\mathbb{R}^{m \times n}$  é unitariamente invariante se, dada uma matriz  $A \in \mathbb{R}^{m \times n}$ ,  $\|Q_1 A Q_2^\top\| = \|A\|$ , para quaisquer matrizes ortogonais  $Q_1 \in \mathbb{R}^{m \times m}$  e  $Q_2 \in \mathbb{R}^{n \times n}$ .

**Definição 6 (Melhores Aproximações Matriciais).** Seja  $r$  o posto de uma matriz não nula  $A$ . Chamamos de melhor aproximação de posto  $k$  para  $A$  no sentido das normas unitariamente invariantes, denotando-a por  $A_k$ , a matriz de posto  $k \leq r$  que é a soma das  $k$  matrizes de posto unitário formadas a partir dos  $k$  primeiros valores e vetores singulares de  $A$ . Isto é,

$$A_k = \sum_{i=1}^k \sigma_i u_i v_i^\top, \quad (2.86)$$

em que  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_k > 0$  são os valores singulares de  $A$  e  $u_1, u_2, \dots, u_r$  e  $v_1, v_2, \dots, v_r$  são seus respectivos vetores singulares à esquerda e à direita. Por simplicidade,  $A_k$  também será chamada de “melhor aproximação para  $A$ ”, sem prejuízo do seu significado.

### 3 MÉTODOS COMPUTACIONAIS

Neste capítulo, apresentamos os dois métodos de otimização que serão adotados nos testes computacionais da parte experimental deste trabalho: o Método dos Multiplicadores com Direções Alternadas (ADMM, de *Alternating Direction Method of Multipliers*), cujos resultados nos servirão de referência, dado o histórico de bom desempenho reportado na literatura, e o que nomeamos de Método do Gradiente Proximal Espectral (SPGM, de *Spectral Proximal Gradient Method*), a ser introduzido em seguida.

Primeiramente, enunciaremos o problema da decomposição posto-esparsidade em que reside nosso interesse, estabelecendo a notação para este e os próximos capítulos.

**Definição 7** (Problema da Decomposição Posto-Esparsidade (SLRMD)). Seja  $A \in \mathbb{R}^{m \times n}$  a matriz formada pela soma de uma matriz  $S^*$  esparsa com uma matriz  $L^*$  de posto baixo cujos padrões de esparsidade e posto não sejam conhecidos. Suponha que as condições de recuperabilidade estabelecidas em Candès et al. ([21]) sejam satisfeitas. Queremos recuperar as matrizes  $S^*$  e  $L^*$  por meio da resolução do seguinte problema de otimização:

$$\begin{aligned} & \underset{S, L \in \mathbb{R}^{m \times n}}{\text{minimizar}} \quad \left[ \Psi(S, L) = t \|\text{vec}(S)\|_1 + (1-t) \|L\|_* \right] \\ & \text{sujeito a} \quad S + L = A, \end{aligned} \quad (3.1)$$

em que  $t$  é a constante dada por

$$t = \frac{1}{1 + \sqrt{\max\{m, n\}}}. \quad (3.2)$$

#### 3.1 Método dos Multiplicadores com Direções Alternadas (ADMM)

A ideia por trás da versão clássica do ADMM é resolver o problema de otimização minimizando seu Lagrangiano aumentado por blocos ([16]), já que idealmente é aplicado a funções objetivo que sejam separáveis em duas outras funções que não compartilhem variáveis entre si. Este é precisamente o caso do Problema (3.1): a norma 1 só opera na variável matricial  $S$  e a norma nuclear, em  $L$ .

O Lagrangiano  $\rho$ -aumentado associado ao problema da decomposição posto-esparsidade tem a seguinte expressão:

$$\mathcal{L}_\rho(S, L, Y) = t \|\text{vec}(S)\|_1 + (1-t) \|L\|_* + \langle Y, S + L - A \rangle + \frac{\rho}{2} \|S + L - A\|_F^2, \quad (3.3)$$

em que  $\rho > 0$  é o parâmetro de “penalização” e  $Y \in \mathbb{R}^{m \times n}$  é a matriz de variáveis duais (ou, ainda, matriz de multiplicadores de Lagrange).

Da teoria de dualidade, sabemos que se  $(\hat{S}, \hat{L}, \hat{Y})$  é um ponto de sela do Lagrangiano aumentado, então  $(\hat{S}, \hat{L})$  é uma solução do problema original (primal) e  $\hat{Y}$  é o multiplicador de Lagrange associado (solução dual) ([56, Capítulo 5]). Portanto, uma estratégia iterativa seria minimizar o Lagrangiano nas variáveis primais e atualizar os multiplicadores na direção de máxima subida com relação às variáveis duais ([82]). Uma dificuldade está na penalização quadrática, que tipicamente dificulta a primeira etapa dessa estratégia ao impedir a separação das variáveis no subproblema de otimização. Assim, a minimização por blocos do ADMM torna-se atrativa por propiciar essa separabilidade, como veremos adiante.

Prosseguindo como em [103], realizamos a minimização do Lagrangiano aumentado por blocos, tirando proveito de sua estrutura separável, isto é:

$$S^{(k+1)} \stackrel{\text{def}}{=} \arg \min_X \mathcal{L}_\rho(X, L^{(k)}, Y^{(k)}) \quad (3.4)$$

$$L^{(k+1)} \stackrel{\text{def}}{=} \arg \min_X \mathcal{L}_\rho(S^{(k+1)}, X, Y^{(k)}) \quad (3.5)$$

$$Y^{(k+1)} \stackrel{\text{def}}{=} Y^{(k)} + \rho (S^{(k+1)} + L^{(k+1)} - A). \quad (3.6)$$

Completando quadrado nas minimizações por bloco, eliminando os termos constantes e escalando a matriz de multiplicadores por um fator  $1/\rho$ , podemos obter um esquema equivalente ao anterior porém em termos dos operadores proximais das normas 1 e nuclear:

$$\begin{aligned} S^{(k+1)} &= \text{prox}_{\left(\frac{t}{\rho} \|\cdot\|_1\right)} \left( A - L^{(k)} - U^{(k)} \right) \\ L^{(k+1)} &= \text{prox}_{\left(\frac{1-t}{\rho} \|\cdot\|_*\right)} \left( A - S^{(k+1)} - U^{(k)} \right) \\ U^{(k+1)} &= U^{(k)} + S^{(k+1)} + L^{(k+1)} - A. \end{aligned} \quad (3.7)$$

Note que esse é o mesmo esquema apresentado por Candès *et al.* ([21]).

A grande conveniência de expressar as minimizações por blocos por meio dos operadores proximais das normas envolvidas é que ambas possuem fórmulas fechadas amplamente conhecidas: os operadores *soft-thresholding*, para a norma 1, e o *singular value thresholding* (veja [77, 78] e [28]).

Uma das dificuldades está na escolha do valor do parâmetro de penalização  $\rho$ , que não é imediata. De fato, é possível penalizar cada restrição com um valor diferente, especialmente quando há particularidades em suas interpretações e formas. Além disso, é possível variar o(s) valor(es) desse(s) parâmetro(s) a cada iteração, o que de fato pode trazer melhores resultados práticos, como reportados, por exemplo, em

([49, 58]). Neste trabalho, assim como em [103] e [21], adotamos um valor fixo para  $\rho$  que é dado em termos dos dados de cada instância, já que se os resultados relatados nesses trabalhos foram satisfatórios. Assim, fazemos com que  $\rho$  valha uma fração do inverso das médias das magnitudes de  $A$ , ou seja,

$$\rho = \frac{1}{4} \frac{m n}{\|\text{vec}(A)\|_1}. \quad (3.8)$$

Os critérios de parada do ADMM se baseiam nas únicas duas condições que precisam ser satisfeitas para que se tenha garantia de otimalidade: factibilidade primal e tamanho do passo suficientemente pequeno. Para um desenvolvimento completo, veja [16, Seção 3.1.1].

Observe que o esquema de atualização em (3.4) segue a “filosofia de Gauss-Seidel” de usar sempre a informação mais atualizada. Com isto, do par inicial de pontos  $(S^{(0)}, L^{(0)})$ , a componente matricial  $S^{(0)}$  é sempre descartada. No entanto, note que a ordem de atualização entre  $S$  e  $L$  na minimização por blocos é arbitrária e, em teoria, poderia ser alterada se necessário. Aqui, mantemos a ordem padrão encontrada na literatura, atualizando primeiro  $S$  e depois  $L$ . O Algoritmo 1 reúne os procedimentos realizados pelo ADMM.

## 3.2 Método do Gradiente Proximal Espectral (SPGM)

Sejam  $f : \mathbb{R}^p \rightarrow \mathbb{R}$  uma função suave com gradiente Lipschitz contínuo e  $c : \mathbb{R}^p \rightarrow \mathbb{R}$  uma função convexa chamada de termo regularizador. Consideremos o problema de minimizar uma combinação dessas duas funções:

$$\underset{x \in \mathbb{R}^p}{\text{minimizar}} \left[ \phi(x) \stackrel{\text{def}}{=} f(x) + \omega c(x) \right] \quad (3.9)$$

para algum  $\omega \in (0, \infty)$ . Chamamos de Método do Gradiente Proximal Espectral (SPGM, do inglês *Spectral Proximal Gradient Method*) ao método que resolve o Problema (3.9) através de sucessivas minimizações de modelos quadráticos locais para  $f(x)$  somados ao termo  $\omega c(x)$ . As hessianas dos modelos quadráticos para  $f$  são matrizes escalares  $\alpha_k I$  construídas de acordo com a estratégia de Barzilai-Borwein [3]. A cada iteração, o algoritmo realiza uma busca para encontrar o iterando seguinte duplicando o valor de  $\alpha_k$  até que a solução do modelo correspondente satisfaça certo critério não monótono de decréscimo suficiente, como é comum nesses casos. Em certos sentidos, nossa abordagem tem semelhanças com os trabalhos [10, 53, 101].

No contexto da decomposição posto-esparsidade, identificaremos  $f$  com a função de suavização de Huber ( $h_\mu$ , com parâmetro  $\mu > 0$ ), definida no final desta seção, aplicada à norma 1 de vetores, ao passo que  $c$  será associada ao termo da norma nuclear de matrizes na função objetivo do Problema (2.7). Dessa forma, os

**Algoritmo 1** – Método dos Multiplicadores com Direções Alternadas (ADMM)

- 1 **entrada:** Matriz  $A$  de entrada; pontos iniciais  $S^{(0)}$  e  $L^{(0)}$  para as componentes, e  $U^{(0)}$  para os multiplicadores de Lagrange; tolerâncias  $\epsilon_{\text{relinf}}$  para a infactibilidade relativa e  $\epsilon_x$  para o passo (variação) relativo no domínio.
- 2 **saída** : Par  $(S^{(k)}, Y^{(k)})$ .
- 3 Calcule o parâmetro  $\rho$  de penalização como uma fração do inverso da magnitude média das entradas de  $A$ :
- 4  $\rho \leftarrow \frac{1}{4} \frac{mn}{\|\text{vec}(A)\|_1}$ ;
- 5 Inicialize o contador de iterações:  $k \leftarrow 0$ ;
- 6 **enquanto**

$$\frac{\|(S^{(k+1)}, L^{(k+1)}) - (S^{(k)}, L^{(k)})\|_F}{1 + \|(S^{(k)}, L^{(k)})\|_F} > \epsilon_x, \quad \frac{\|S^{(k+1)} + L^{(k+1)} - A\|_F}{\|A\|_F} > \epsilon_{\text{relinf}}, \quad k < M_{\text{it}},$$

**faça**

- 7  $S^{(k+1)} \leftarrow \text{prox}_{\left(\frac{t}{\rho} \|\cdot\|_1\right)}(A - L^{(k)} - U^{(k)});$
- 8  $L^{(k+1)} \leftarrow \text{prox}_{\left(\frac{1-t}{\rho} \|\cdot\|_*\right)}(A - S^{(k+1)} - U^{(k)});$
- 9  $U^{(k+1)} \leftarrow U^{(k)} + (S^{(k+1)} + L^{(k+1)} - A);$
- 10  $k \leftarrow k + 1;$

11 **fim**

- 12 **retorne**  $(S^{(k)}, L^{(k)})$

minimizadores dos modelos que aproximam  $\phi$  a cada iteração podem ser calculados através de fórmulas fechadas que envolvem essencialmente a aplicação do operador proximal da norma nuclear – o chamado *Singular Value Thresholding* ([19]) – a passos na direção de máxima descida para  $f$ .

Além disso, acoplaremos ao SPGM um esquema de continuação homotópica aplicado ao parâmetro  $\mu$  de suavização, que será reduzido gradativamente ao longo do processo de otimização. A ideia é que o método seja iniciado com um valor de  $\mu$  para o qual a função resultante seja mais *bem comportada* e mais fácil de ser minimizada. Dessa forma, espera-se que com poucas iterações o método consiga encontrar soluções aproximadas para esse subproblema suavizado. Em seguida, o processo é completamente reiniciado a partir do ponto encontrado na etapa anterior, e assim sucessivamente até que se atinja a precisão desejada, atrelada ao valor final de  $\mu$ .

Note que, na prática, temos uma sucessão de problemas suavizados regidos pela sequência decrescente  $(\mu_i)$ . Conforme  $\mu_i$  tende a zero, a função suavizada  $h_{\mu_i}$  tende

à função  $f$ . Além disso, para cada  $\mu_i$ , o SPGM executa sua estratégia de minimização por meio dos modelos acima mencionados. Portanto, temos uma sequência de *iterações homotópicas* atrelada à sequência  $(\mu_i)$  e, para cada iteração homotópica, temos uma sequência de subproblemas resolvidos por iterações que chamamos de *iterações ordinárias*. Por fim, vale lembrar uma outra sequência de procedimentos encaixada nas iterações ordinárias, como mencionamos: a busca linear no tamanho do passo espectral para a função de Huber. Detalhamos essas etapas no restante deste capítulo.

### 3.2.1 O modelo local para a função objetivo

Cada iteração  $k$  do SPGM corresponde à minimização de um modelo  $Q_k$  para a função objetivo  $\phi$  do Problema (3.9) centrado no iterando  $x^{(k)}$ :

$$\underset{z}{\text{minimizar}} \left[ Q_k(z) \stackrel{\text{def}}{=} q_k(z) + \omega c(z) \right], \quad (3.10)$$

em que

$$q_k(z) \stackrel{\text{def}}{=} f(x^{(k)}) + \nabla f(x^{(k)})^\top (z - x^{(k)}) + \frac{\alpha_k}{2} \|z - x^{(k)}\|_2^2 \quad (3.11)$$

é um modelo quadrático para  $f$  no qual o termo  $\alpha_k I$  pode ser entendido como uma aproximação para a matriz hessiana  $\nabla^2 f(x^{(k)})$ , quando esta existe.

Em cada iteração  $k$ , o valor  $\alpha_k$  é escolhido de forma que  $\alpha_k I$  seja a matriz escalar que melhor resolve a equação secante no sentido de quadrados mínimos, como proposto por Barzilai e Borwein [3] para dimensão 2 e demonstrado para o caso geral  $n$ -dimensional por Raydan [83]. Em outras palavras, a primeira tentativa em cada iteração é escolher  $\alpha_k$  como o escalar que minimiza o resíduo da equação secante na norma 2, isto é,

$$\alpha_+ = \arg \min_{\alpha} \left\| \alpha s^{(k-1)} - y^{(k-1)} \right\|^2 = \frac{(s^{(k-1)})^\top y^{(k-1)}}{(s^{(k-1)})^\top s^{(k-1)}}, \quad (3.12)$$

em que  $s^{(k-1)} = x^{(k)} - x^{(k-1)}$  e  $y^{(k-1)} = \nabla f(x^{(k)}) - \nabla f(x^{(k-1)})$ .

Caso o ponto tentativo resultante desse passo espectral não satisfaça um critério de decréscimo suficiente não monótono, duplicamos<sup>1</sup> o valor de  $\alpha_k$  e recalculamos o ponto tentativo. Esse processo de busca é repetido até que o critério de aceite seja satisfeito e o ponto associado seja declarado como iterando seguinte, ou até que um número máximo de tentativas na busca seja realizado.

Na prática, para garantir que  $\alpha_k$  não fique muito pequeno nem muito grande, seu valor é calculado de maneira que  $\alpha_k \in [\alpha_{\min}, \alpha_{\max}]$ , com constantes  $\alpha_{\min}$  e  $\alpha_{\max}$  tais que  $0 < \alpha_{\min} < \alpha_{\max}$ . Assim,

$$\alpha_k = \min\{\alpha_{\max}, \max\{\alpha_{\min}, \alpha_+\}\}. \quad (3.13)$$

<sup>1</sup> Como veremos na Equação (3.17), aumentar  $\alpha_k$  implica em diminuir o tamanho do passo do tipo gradiente.

Podemos reescrever o subproblema (3.10) completando os termos quadráticos e eliminando constantes indesejadas, de tal maneira que as seguintes equivalências podem ser estabelecidas:

$$\begin{aligned} & \arg \min_z Q_k(z) \\ &= \arg \min_z \omega c(z) + \frac{1}{2\alpha_k} \left\| \nabla f(x^{(k)}) \right\|^2 + \nabla f(x^{(k)})^\top (z - x^{(k)}) + \frac{\alpha_k}{2} \left\| z - x^{(k)} \right\|^2 \end{aligned} \quad (3.14)$$

$$= \arg \min_z \omega c(z) + \frac{\alpha_k}{2} \left\| z - u^{(k)} \right\|^2 \quad (3.15)$$

$$= \arg \min_z c(z) + \frac{\alpha_k}{2\omega} \left\| z - u^{(k)} \right\|^2, \quad (3.16)$$

em que

$$u^{(k)} = x^{(k)} - \frac{1}{\alpha_k} \nabla f(x^{(k)}). \quad (3.17)$$

O Problema (3.16) é precisamente a definição do operador proximal da função  $c$  escalada pelo fator  $\frac{\omega}{\alpha_k}$  e avaliado em  $u^{(k)}$  (ver [28, Equação (2.13)]. Ou seja, podemos escrever explicitamente o ponto tentativo  $x_+^{(k)}$  da iteração  $k$  associado ao passo tentativo espectral  $1/\alpha_k$  como

$$x_+^{(k)} = \arg \min_z Q_k(z) = \text{prox}_{\frac{\omega}{\alpha_k} c} \left( x^{(k)} - \frac{1}{\alpha_k} \nabla f(x^{(k)}) \right). \quad (3.18)$$

Daí o termo “proximal” na nomenclatura do método.

Claramente, a aplicação desse esquema algorítmico é atrativa apenas nos casos em que a resolução computacional do subproblema possa ser concretizada de maneira *eficiente*. Em nosso contexto, em que  $f$  é a função de suavização de Huber e  $c$  é a norma nuclear, a resolução do subproblema possui fórmula explícita de resolução.

### 3.2.2 Critério não monótono de decréscimo suficiente

O uso de critérios não monótonos de aceite durante a etapa de busca linear em esquemas algorítmicos globalmente convergentes que incorporam o passo espectral de Barzilai-Borwein é comum na literatura ([46, 53, 101]). A ideia é requerer que o valor objetivo seja ligeiramente menor do que o maior valor objetivo alcançado nas  $M$  iterações prévias, sendo  $M$  um número natural fixo.

Conforme relatado em [101, Seção II.F], apesar da ocorrência de grandes aumentos ocasionais no valor da função objetivo, permitir que eles ocorram parece ser uma característica essencial para o bom desempenho da otimização. O critério de aceite adotado neste trabalho segue essas ideias.

Na iteração  $k$ , um ponto  $x_+^{(k+1)}$  candidato a iterando seguinte (ponto tentativo) será considerado aceite se o valor  $\phi(x_+^{(k+1)})$  da função objetivo em  $x_+^{(k+1)}$  for um

pouco menor do que o maior valor de  $\phi$  nas últimas  $M + 1$  iterações, isto é, se

$$\phi(x_+^{(k+1)}) \leq \left( \max_{i=\max(k-M,0),\dots,k} \phi(x^{(i)}) \right) - \frac{\sigma}{2} \alpha_+ \|x_+^{(k+1)} - x^{(k)}\|^2, \quad (3.19)$$

em que  $\sigma \in (0, 1)$  é uma constante tipicamente próxima a zero. Note que, quanto maior o tamanho do passo imposto pelo ponto candidato, mais exigente o critério de decréscimo se torna. Por outro lado, quando o ponto candidato é mais próximo ao iterando corrente, a exigência é mais branda.

### 3.2.3 Suavização da norma 1 e operador SVT

Para adaptar o nosso problema aos moldes de (3.9), suavizaremos a norma 1 presente na função objetivo do problema convexo não diferenciável (2.7) e o convertemos em um problema irrestrito através da substituição  $S = A - L$ . Ou seja, trocaremos o problema

$$\underset{X \in \mathbb{R}^{m \times n}}{\text{minimizar}} \quad t \|\text{vec}(X - A)\|_1 + (1 - t) \|X\|_*$$

por

$$\underset{X \in \mathbb{R}^{m \times n}}{\text{minimizar}} \quad t h_\mu(X - A) + (1 - t) \|X\|_* \quad (3.20)$$

em que  $h_\mu$  é a função de Huber multivariável, convexa e suave, definida em (2.71). Com isto, resta apenas explicitar o operador proximal da norma nuclear, conhecido como *Singular Value Thresholding* (SVT) (cf. [19]):

$$\text{prox}_{\lambda \|\cdot\|_*}(X) = \sum_{i=1}^r [\sigma_i - \lambda]_+ u_i v_i^\top, \quad (3.21)$$

em que

$$X = \sum_{i=1}^r \sigma_i u_i v_i^\top \quad (3.22)$$

é a forma reduzida da decomposição em valores singulares (SVD) da matriz  $X$ . Reforçamos que o foco deste trabalho está na suavização da norma 1 – englobada pelo termo  $f(x)$  em (3.9) – e na preservação da norma nuclear – correspondente a  $c(x)$  na mesma equação. Por isso destacamos apenas o operador da norma nuclear, e não o da norma 1, o qual só apareceria se suavizássemos a norma nuclear em vez da norma 1.

## 3.3 Método do Gradiente Proximal Espectral: Algoritmos

### 3.3.1 Formulação do Problema

Vimos a ideia do funcionamento do SPGM aplicado ao Problema (3.9) em dimensão  $n$ . Explicitemos, agora, as estruturas análogas para tratar de variáveis

matriciais na situação específica em que uma das funções que compõem a função objetivo foi suavizada.

O SPGM foi pensado para resolver problemas que possam ser escritos da seguinte maneira:

$$\underset{X \in \mathbb{R}^{m \times n}}{\text{minimizar}} \left[ \Psi(X; \mu) \stackrel{\text{def}}{=} \Psi(X) = s_\mu(X - A) + c(X) \right],$$

em que  $s_\mu$  é uma função suave com parâmetro  $\mu$  e  $c$  é uma função contínua convexa não necessariamente suave e cujo operador proximal é *razoavelmente* fácil de calcular. Na definição de  $\Psi$  há uma dependência implícita do parâmetro de suavização  $\mu$ , porém a omitiremos para não carregar a notação.

Nosso interesse reside principalmente nos casos em que  $c$  é a norma 1 ou a norma nuclear e, como vimos, qualquer uma delas pode ser suavizada usando as técnicas vistas na seção anterior. Para obter problemas irrestritos, isolamos uma das componentes na restrição de igualdade e a substituímos na função objetivo para obter

$$\underset{L \in \mathbb{R}^{m \times n}}{\text{minimizar}} \quad \tau h_\mu(L - A) + \|L\|_\star \quad \text{ou} \quad \underset{S \in \mathbb{R}^{m \times n}}{\text{minimizar}} \quad h_\mu(S - A) + \tau \|S\|_1.$$

Observe que a variável em termos da qual o problema é escrito é arbitrária: em todos os casos podemos optar por  $S$  ou  $L$ . Entretanto, decidimos manter a variável cujas propriedades dependem do termo que não está sendo suavizado.

Chamemos de  $X \in \mathbb{R}^{m \times n}$  a variável do problema irrestrito. Na tentativa de resolver  $\min_X \Psi(X)$ , realizaremos sucessivas minimizações de aproximações para  $\Psi$  usando modelos quadráticos para  $s_\mu$ . A cada iteração  $k$ , o modelo será escrito como a soma da aproximação de primeira ordem centrada em  $X^{(k)}$  com a informação de segunda ordem:

$$\underset{X \in \mathbb{R}^{m \times n}}{\text{minimizar}} \left[ Q_k(X) = q_k(X) + \omega c(X) \right], \quad (3.23)$$

$$q_k(X) = s_\mu(X^{(k)} - A) + \left\langle \nabla s_\mu(X^{(k)} - A), X - X^{(k)} \right\rangle + \frac{\alpha_k}{2} \|X - X^{(k)}\|_F^2. \quad (3.24)$$

Manipulando algebricamente para completar quadrados, podemos reescrever esse modelo como a soma de um termo constante com relação a  $X$  e o quadrado de um termo envolvendo  $X$  (as contas em detalhe foram disponibilizadas para consulta no Apêndice A):

$$q_k(X) = s_\mu(X^{(k)} - A) + \left\langle \nabla s_\mu(X^{(k)} - A), X - X^{(k)} \right\rangle + \frac{\alpha_k}{2} \|X - X^{(k)}\|_F^2 \quad (3.25)$$

$$= s_\mu(X^{(k)} - A) - \frac{1}{2\alpha_k} \left\| \nabla s_\mu(X^{(k)} - A) \right\|_F^2 + \frac{\alpha_k}{2} \left\| X - X^{(k)} + \frac{1}{\alpha_k} \nabla s_\mu(X^{(k)} - A) \right\|_F^2. \quad (3.26)$$

Portanto, descartando as constantes com relação a  $X$ , somos capazes de escrever a solução do problema de otimização como o resultado da aplicação de um operador proximal

$$\arg \min_X \{q_k(X) + c(X)\} = \text{prox}_{\frac{1}{\alpha_k}c} \left( X^{(k)} - \frac{1}{\alpha_k} \nabla s_\mu(X^{(k)} - A) \right) \quad (3.27)$$

e, assim, destacamos a atualização do SPGM na versão matricial:

$$X^{(k+1)} = \text{prox}_{\frac{1}{\alpha_k}c} \left( X^{(k)} - \frac{1}{\alpha_k} \nabla s_\mu(X^{(k)} - A) \right). \quad (3.28)$$

Note que, se tivéssemos o problema

$$\underset{X \in \mathbb{R}^{m \times n}}{\text{minimizar}} \quad \Psi(X) = \omega_1 s_\mu(X - A) + \omega_2 c(X). \quad (3.29)$$

então, a iteração correspondente seria

$$X^{(k+1)} = \text{prox}_{\frac{\omega_2}{\alpha_k}c} \left( X^{(k)} - \frac{\omega_1}{\alpha_k} \nabla s_\mu(X^{(k)} - A) \right) \quad (3.30)$$

e o escalar de Barzilai-Borwein também seria influenciado pelo fator  $\omega_1$ .

Por fim, se  $c(X) = \|X\|_*$  e  $s_\mu$  corresponde a uma suavização para a norma 1, então

$$\text{prox}_{\frac{1}{\alpha_k}c} \left( X^{(k)} - \frac{1}{\alpha_k} \nabla s_\mu(X^{(k)} - A) \right) = U \left[ \Sigma - \frac{1}{\alpha_k} I \right]_+ V^\top,$$

em que o operador  $[a]_+ \stackrel{\text{def}}{=} \max\{0, a\}$  opera elemento a elemento e

$$X^{(k)} - \frac{1}{\alpha_k} \nabla s_\mu(X^{(k)} - A) = U \Sigma V^\top.$$

Por outro lado, se  $c(X) = \tau \|X\|_1$  e  $s_\mu$  corresponde a uma suavização para a norma nuclear, então

$$\text{prox}_{\frac{1}{\alpha_k}c} \left( X^{(k)} - \frac{1}{\alpha_k} \nabla s_\mu(X^{(k)} - A) \right) = \left[ X^{(k)} - \frac{1}{\alpha_k} \nabla s_\mu(X^{(k)} - A) - \frac{\tau}{\alpha_k} I \right]_+.$$

Observe que a expressão para o operador proximal

De acordo com Beck e Teboulle [5], a reformulação do problema que prepara a aplicação do método proximal é um caso particular do esquema introduzido em [79, modelo (BF), p. 384] para a resolução do problema de minimização de uma função com regularização na norma 1. A demonstração de convergência do ISTA [5] a um minimizador depende do parâmetro “proximal”, que deve ser limitado ao inverso do valor singular de  $A^\top A$  no caso clássico dos quadrados mínimos com regularização via norma 1, e trata-se de um caso particular do demonstrado no Teorema 12.4.6 de [32]. Para casos semelhantes de Algoritmos de Separação (*Splitting Methods*), veja [32, Capítulo 12].

Na Seção 3.3.2, a seguir, constam os algoritmos do SPGM sem e com a homotopia, a qual em geral será adotada em nossos experimentos, e um comentário sobre o aproveitamento da decomposição já feita, no caso da suavização da norma 1.

## 3.3.2 Algoritmos

**Algoritmo 2** – Método do Gradiente Proximal Espectral (SPGM)

- 1 **entrada:** Ponto inicial  $X^{(0)} \in \mathbb{R}^{m \times n}$ ; parâmetro  $\mu$  de suavização, fator  $\eta > 1$  e limitantes  $\alpha_{\max} > \alpha_{\min} > 0$  para o cálculo do tamanho do passo, constantes  $M \in \mathbb{N}$  e  $\xi \in (0, 1)$  do critério de busca linear não monótona, e tolerâncias  $\epsilon_x, \epsilon_f, M_{\text{svd}}$  e  $M_{\text{it}}$ .
- 2 **saída** : Matriz  $X^{(k)}$  e gastos em SVD e iterações.
- 3 Inicialize o contador de iterações:  $k \leftarrow 0$ ;

4 **enquanto**

$$\frac{\|X^{(k)} - X^{(k-1)}\|_F}{1 + \|X^{(k-1)}\|_F} \geq \epsilon_x, \quad \frac{|\Psi(X^{(k)}) - \Psi(X^{(k-1)})|}{1 + |\Psi(X^{(k-1)})|} \geq \epsilon_f, \quad k < M_{\text{it}}.$$

**faça**

5 Determine o parâmetro  $\alpha_k$  relacionado ao tamanho do passo calculando

$$\alpha_k = \begin{cases} 10^{10}, & \text{se } k = 0, \\ \min(\alpha_{\max}, \max(\alpha_{\min}, \alpha_k^+)), & \text{se } k = 1, 2, 3 \dots \end{cases}$$

7 em que  $\alpha_k^+$  é o escalar de Barzilai-Borwein dado por

$$\alpha_k^+ = \begin{cases} \langle R^{(k)}, Y^{(k)} \rangle / \langle R^{(k)}, R^{(k)} \rangle, & \text{if } \langle R^{(k)}, Y^{(k)} \rangle > 0, \\ \alpha_{k-1} / 2 & \text{if } \langle R^{(k)}, Y^{(k)} \rangle \leq 0 \end{cases}$$

9 com  $R^{(k)} = X^{(k)} - X^{(k-1)}$  e  $Y^{(k)} = \nabla_{s_\mu}(X^{(k)} - A) - \nabla_{s_\mu}(X^{(k-1)} - A)$ ;

10 **enquanto**  $\Psi(X^{(k+1)}) > \Psi_{\text{thresh}}$  **faça**

$$11 \quad X^{(k+1)} \leftarrow \text{prox}_{\frac{1}{\alpha_k} c} \left( X^{(k)} - \frac{1}{\alpha_k} \nabla_{s_\mu}(X^{(k)} - A) \right);$$

$$12 \quad \Psi_{\text{thresh}} \leftarrow \max_{i=0, \dots, \min(k, M-1)} \left\{ \Psi(X^{(k-i)}) \right\} - \frac{\xi}{2} \alpha_k \|X^{(k+1)} - X^{(k)}\|_F^2;$$

$$13 \quad \alpha_k \leftarrow \eta \alpha_k;$$

14 **fim**

$$15 \quad k \leftarrow k + 1;$$

16 **fim**

17 **retorne**  $X^{(k)}$ , o número de iterações e o de SVDs.

**Algoritmo 3** – SPGM com Continuação Homotópica

---

```

1 entrada: Ponto inicial  $X^{(0)} \in \mathbb{R}^{m \times n}$ ; sequência decrescente de parâmetros
    $(\mu_{(i)})_{i=1}^T$  de suavização, fator  $\eta > 1$  e limitantes  $\alpha_{\max} > \alpha_{\min} > 0$  para o
   cálculo do tamanho do passo, constantes  $M \in \mathbb{N}$  e  $\xi \in (0, 1)$  do critério
   de busca linear não monótona, e tolerâncias  $\epsilon_x$ ,  $\epsilon_f$ ,  $M_{\text{svd}}$ , e  $M_{\text{it}}$ .
2 saída : Matriz  $X^{(i)}$  e gastos em iterações ( $i$ , nIter) e nSVD.
3 % Iniciar contadores
4 nIter  $\leftarrow 0$ ;
5 nSVD  $\leftarrow 0$ ;
6  $i \leftarrow 1$ ;
7 enquanto  $i \leq T$ , nIter  $< M_{\text{it}}$  e nSVD  $< M_{\text{svd}}$ , faça
8    $\mu \leftarrow \mu_{(i)}$ ;
9    $Z^0 \leftarrow X^{(i-1)}$ ;
10  % Chama o SPGM (Alg. 2) com as devidas tolerâncias
11   $X^{(i)}$ , nItOrd, nSVDOrd  $\leftarrow \text{spgm}(Z^0, \mu, \eta, \alpha_{\max}, \alpha_{\min}, M, \xi, \epsilon_x, \epsilon_f, M_{\text{it}})$ ;
12  % Atualizar parâmetros e tolerâncias
13  nIter  $\leftarrow$  nIter + nItOrd;
14  nSVD  $\leftarrow$  nSVD + nSVDOrd;
15   $M_{\text{it}} \leftarrow M_{\text{it}} - \text{nIter}$ ;
16   $M_{\text{svd}} \leftarrow M_{\text{svd}} - \text{nSVD}$ ;
17   $i \leftarrow i + 1$ ;
18 fim
19 retorne  $X^{(i)}$ , nIter, nSVD

```

---

## Economizando SVDs

Estritamente falando, quando suavizamos a norma 1 e mantemos o termo da norma nuclear, a cada iteração o SPGM precisa calcular o valor da função objetivo e ainda avaliar o operador proximal da norma nuclear, o que a princípio custa duas decomposições em valores singulares – uma por avaliação. Como veremos agora, o custo total por iteração pode ser reduzido a um se aproveitarmos a avaliação do operador proximal.

Se  $W = \text{prox}_{t\|\cdot\|_*}(Z)$  e  $\text{prox}_{t\|\cdot\|_*}(Z) = U[\Sigma - tI]_+V^\top$ , com  $Z = U\Sigma V^\top$ , então  $W = U[\Sigma - tI]_+V^\top$  é uma decomposição em valores singulares para  $W$ . Portanto, o próprio cálculo de  $W$  nos fornece  $\|W\|_*$  de graça, pois  $\|W\|_* = \|\Sigma - tI\|_+$ .

## Parte III

# Experimentos Computacionais

## INTRODUÇÃO À PARTE III

Nesta terceira parte do trabalho, descrevemos os experimentos computacionais realizados e apresentamos os resultados numéricos que sustentam as conclusões deste trabalho.

Inicialmente, introduzimos no Capítulo 4 os problemas-testes com dados artificiais aleatórios, estabelecemos uma configuração padrão para o SPGM e validamos nossa implementação do ADMM para, em seguida, comparar o desempenho dos dois métodos sob condições equiparáveis às dos experimentos reportados na literatura de referência.

O Capítulo 5 traz fundamentos de Processamento de Imagens Digitais que embasam a consideração do que chamamos de Espaço de Exibição e fundamentam a estimativa para a Precisão Alvo a ser almejada nos problemas cujas saídas consistem em imagens, como é o caso das instâncias de videovigilância que descrevemos no Capítulo 6. Analisamos, então, as características de posto e esparsidade que distinguem os problemas teóricos dos práticos que adotamos aqui, cujos aspectos são favoráveis ao SPGM.

Em seguida, encontra-se no Capítulo 7 o cerne do trabalho, em que o SPGM confirma nossas conjecturas e se coloca como uma alternativa mais eficiente e robusta ao ADMM na classe de problemas e na precisão que admitimos.

Por fim, no Capítulo 8, elaboramos e incorporamos uma heurística de escolha do ponto inicial que, surpreendentemente, impulsiona mais o ADMM do que o SPGM, de forma que o primeiro se coloca como pário ao segundo, também em eficiência e robustez. Encerramos o trabalho discutindo aspectos visuais das Soluções de Referência, bem como suas limitações, e estipulamos direções de investigação visando contornar as dificuldades da formulação do problema quanto a da calibração dos parâmetros do SPGM.

Os experimentos computacionais deste trabalho foram executados em um microcomputador com processador Intel<sup>®</sup> Core<sup>™</sup> i7-8700 CPU @ 3.20GHz, 6 núcleos e 12 processadores lógicos com base em x64, memória instalada (RAM) de 32GB e sistema operacional *Ubuntu* 20.04 LTS. O ambiente de computação numérica utilizado foi o MATLAB<sup>®</sup> R2019a (9.6). A preparação de gráficos, tabelas e do próprio texto foi realizada majoritariamente em um *laptop* Dell Inspiron 5557 com processador Intel<sup>®</sup> Core<sup>™</sup> i7-6500U CPU @ 2.50GHz, 2 núcleos e 4 processadores lógicos com base em x64, memória instalada (RAM) de 16 GB e sistema operacional *Microsoft Windows 10 Home Single Language* de 64 bits. A versão do MATLAB<sup>®</sup>, nesse caso, foi a R2016a (9.0).

## 4 EXPERIMENTOS NUMÉRICOS COM DADOS ARTIFICIAIS ALEATÓRIOS

Neste Capítulo, realizamos experimentos computacionais comparando o desempenho dos métodos ADMM e SPGM em problemas-testes cujas matrizes de entrada são aleatórias e satisfazem condições teóricas que asseguram a decomposição matricial por meio da resolução do problema SLRMD. Trata-se da primeira parte de experimentos computacionais que, em última instância, visam o estudo do desempenho dos métodos em problemas de otimização com dados advindos de situações práticas de aplicação.

Primeiramente, introduzimos as diretrizes de construção das instâncias aleatórias, baseadas nos trabalhos de Candès *et al.* ([21]) e Yuang e Yang ([103]), para, logo depois, validar nossa implementação do ADMM por meio da comparação dos nossos resultados com aqueles publicados nos trabalhos mencionados. Em seguida, desenvolvemos uma análise estrutural das instâncias aleatórias e verificamos a qualidade de algumas soluções triviais para os problemas gerados. Por fim, discutimos algumas configurações para o SPGM e comparamos os resultados obtidos com ambos métodos, propondo uma reflexão a respeito da representatividade das instâncias aleatórias com relação às de aplicação.

### 4.1 Algumas Definições, Notações e Nomenclaturas Preliminares

Nas tabelas e figuras da parte experimental deste trabalho, utilizamos algumas notações e nomenclaturas com o fim de facilitar a exposição dos dados, seja nos textos ou nos gráficos. Por motivos de diagramação, às vezes adotamos notações diferentes para o que está no texto, em modo matemático, e o que aparece nas tabelas e gráficos. Os acrônimos, de um modo geral, foram pensados para favorecer a compreensão, embora por diversas vezes sua origem esteja no termo correspondente em inglês, razão pela qual descrevemos brevemente os principais deles a seguir.

#### Representação Numérica

Sobre a representação numérica, vale lembrar que adotamos o ponto (“.”) como separador decimal no lugar da vírgula, por facilidade na reprodução dos resultados computacionais. Pelo mesmo motivo, em diversos momentos usamos a notação exponencial do MATLAB no lugar da notação científica. A título de ilustração, com essas convenções, “1.23e-04” corresponde a  $1.23 \times 10^{-4}$ .

## Métodos Computacionais

Os nomes dos métodos computacionais são os mesmos de antes: ADMM e SPGM. Por questão de espaço e legibilidade, em alguns gráficos e tabelas aparece apenas a primeira letra da sigla, caso em que seu significado fica implícito pelo contexto ou explícito pelas legendas. Além disso, em determinados momentos admitimos configurações distintas para um mesmo método, o que está devidamente evidenciado nas legendas.

## Erros Relativos

Uma medida de importância central em nossas análises é o erro relativo. É por intermédio do erro que avaliamos a qualidade dos iterandos. A Definição 8 expressa uma fórmula geral para o cálculo dessa medida. Entendendo que a solução do problema original é o par de matrizes  $(S, L)$ , chamamos de *erro relativo total* aquele calculado com relação à solução  $(S^*, L^*)$ , o qual foi denotado por RelErr ou  $ER(S, L)$ . Os erros relativos por componente matricial são indicados por  $ER(S)$  (ou ErrSP) e  $ER(L)$  (ou ErrLR).

**Definição 8 (Erro Relativo (ER)).** Considere duas matrizes  $X$  e  $X^*$  em  $\mathbb{R}^{m \times n}$  e suponha que  $X$  é uma aproximação para a matriz de referência  $X^*$  não nula. A fim de compará-las, definimos o erro relativo como o quociente entre a norma do erro matricial e a norma da matriz de referência, ambas normas de Frobenius, isto é:

$$ER(X; X^*) = \frac{\|X - X^*\|_F}{\|X^*\|_F}. \quad (4.1)$$

Na notação acima, os argumentos  $X$  e  $X^*$  podem ser substituídos pelas matrizes-componentes –  $S$  e  $S^*$ , ou  $L$  e  $L^*$  – ou, ainda, pelas matrizes  $(S \ L)$  e  $(S^* \ L^*)$ , respectivamente. Essas especificações nos argumentos podem ser omitidas quando forem dedutíveis pelo contexto ou simplificadas com rótulos correspondentes, como “-S”, “-L” e “-SL”.

## Tamanho do Passo

Outra medida com papel fundamental, em particular para o teste de parada do SPGM, é o tamanho relativizado do passo no domínio ou, alternativamente, a variação relativa no tamanho do passo, a qual chamamos de RelStep e cuja expressão consta na Definição 9. Assim como no caso dos erros, podemos nos referir ao tamanho do passo com relação a uma das componentes ou à justaposição de ambas. Usualmente, no cálculo de  $RelStep(Y; X)$  na iteração  $k$ , adotamos  $X = L^{(k-1)}$  e  $Y = L^{(k)}$ , no SPGM, ao passo que, no ADMM, utilizamos  $X = (S^{(k-1)}, L^{(k-1)})$  e  $Y = (S^{(k)}, L^{(k)})$ , embora façamos referência apenas a “RelStep” em cada caso, sem prejuízo da compreensão.

**Definição 9 (Variação Relativa no Tamanho do Passo).** No contexto de uma sequência de pontos gerados por um método iterativo de otimização, considere dois iterandos  $X$  e  $Y$  no espaço normado  $\mathbb{R}^{m \times n}$  e suponha que  $X$  antecede  $Y$  nessa sequência. Chamamos de tamanho absoluto do passo de  $X$  para  $Y$  à distância entre essas matrizes, medida na norma de Frobenius, ou seja,  $\|Y - X\|_F$ . Definimos também o tamanho relativo do passo de  $X$  para  $Y$  como a razão entre o tamanho do passo e a norma do ponto de partida, isto é:

$$\text{RelStep}(Y; X) = \frac{\|Y - X\|_F}{1 + \|X\|_F}. \quad (4.2)$$

Na notação acima, os argumentos  $X$  e  $Y$  podem ser substituídos por iterandos de uma das matrizes-componentes ( $S^{(j)}$  e  $S^{(k)}$ , ou  $L^{(j)}$  e  $L^{(k)}$ ) ou, ainda, de ambas, com  $(S^{(j)} L^{(j)})$  e  $(S^{(k)} L^{(k)})$ . Essas especificações nos argumentos podem ser omitidas quando foram dedutíveis pelo contexto ou simplificadas com rótulos correspondentes, como “-S”, “-L” e “-SL”.

### Esparsidade e posto aproximados

No cálculo da taxa de esparsidade (*spr*), quando usamos alguma tolerância  $\epsilon > 0$  para considerar como nulos os elementos cuja magnitude é menor do que  $\epsilon$ , indicamos por “ $\approx\text{spr}$ ” a taxa de esparsidade aproximada. De forma análoga, para o cálculo do posto percentual (*rkr*), quando usamos uma tolerância maior do que o padrão do MATLAB<sup>1</sup>, adotamos a designação “ $\approx\text{rnk}$ ”.

### Valor objetivo

Chamamos a atenção para o entendimento e a referência ao termo “função objetivo” devido à existência de três problemas de otimização a que nos referimos frequentemente. De um modo geral, *ObjV* se refere ao valor ótimo do problema que o *solver* em questão é colocado para resolver. Isto significa que *ObjV* para o ADMM é calculado avaliando a função

$$\Psi(S, L) = t \|\text{vec}(S)\|_1 + (1 - t) \|L\|_* \quad (4.3)$$

na aproximação  $(S, L)$  obtida. Por outro lado, para o SPGM, que admite como variáveis apenas as entradas da componente  $L$  e que tem como função objetivo a suavização parcial da função usada pelo ADMM, consideramos como *ObjV* o valor

$$\Psi_\mu(L) = t \left[ \sum_{i,j=1}^n h_\mu(\ell_{ij} - a_{ij}) \right] + (1 - t) \|L\|_*, \quad (4.4)$$

<sup>1</sup> Por *default*, o comando `rank` do MATLAB utiliza a tolerância  $\max(m, n) \cdot \epsilon(\sigma_1(A))$  para determinar o posto de uma matriz  $A \in \mathbb{R}^{m \times n}$ , em que  $\sigma_1$  é o maior valor singular dessa matriz e  $\epsilon(x)$  é a distância entre  $|x|$  e o menor número em ponto flutuante maior que  $|x|$ .

em que  $L = (\ell_{ij}) \in \mathbb{R}^{m \times n}$  e  $h_\mu$  é a função de suavização de Huber com parâmetro  $\mu$ , definida em (2.70).

Dessa forma, para dar sentido às comparações entre os valores objetivos, utilizamos o termo `OrigObjV` para fazer referência ao valor da função objetivo do problema convexo não diferenciável - a *função objetivo original* - avaliada a partir da viabilização do ponto  $L$ , isto é,

$$\Psi(L - A, L) = t \|\text{vec}(L - A)\|_1 + (1 - t) \|L\|_\star. \quad (4.5)$$

### Parâmetro de Compromisso

Nas tabelas com coluna de nome  $t$ , o parâmetro de compromisso entre o posto e esparsidade, exibimos seu valor em porcentagem, já que está restrito ao intervalo  $(0, 1)$ . Sendo assim, o que de fato se exhibe nessas colunas é o valor  $100t$  com arredondamento em caso de necessidade, como geralmente ocorre.

### Tempo

Embora nossa unidade de custo seja o número de SVDs, optamos por exibir algumas medições de tempo. Por influência dos núcleos de processamento do computador, medimos tanto o tempo decorrido (*wall-clock time*), sob o nome `Duração`, quanto o tempo de CPU (que soma os tempos dos núcleos e é sempre maior ou igual ao tempo decorrido), sob o nome `CPU`. Ocasionalmente, esses tempos podem incluir não apenas a execução do método mas também todo o processamento e armazenamento de dados que ocorre ao longo das iterações (procedimento que varia inclusive do ADMM para o SPGM) e, portanto, devem ser vistos com cautela, não servindo de comparação com os valores de outros trabalhos.

### Motivo de Parada

Em diversas tabelas, a coluna `Parada` exhibe por meio dos termos abaixo o motivo pelo qual o algoritmo terminou sua execução:

- **iter**: por número de iterações;
- **relstep**: pelo tamanho relativizado do passo no domínio (`RelStep`) ser menor do que uma tolerância preestabelecida.
- **relinfeas** (ADMM): pelo tamanho da infactibilidade primal relativa,

$$\left\| A - S^{(k)} - L^{(k)} \right\|_F / \|A\|_F,$$

que ficou abaixo de alguma tolerância, indicando factibilidade numérica.

Podemos ter, por fim, a coluna **nIter** representando o número de iterações e a **nSVD**, com a quantidade de decomposições em valores singulares realizadas.

### *Data Profiles*

Com a finalidade de comparar o desempenho dos dois métodos sob investigação, utilizamos gráficos que indicam o percentual de problemas *resolvidos* por cada um dos métodos (eixo vertical) em termos do número de SVDs realizadas durante sua execução (eixo horizontal). Esse tipo de gráfico foi batizado de *data profile* por Moré e Wild [70, Seção 2.2] em seu artigo sobre a comparação de desempenho de algoritmos em Otimização Sem Derivadas (DFO, do inglês *Derivative Free Optimization*).

Assim como no contexto de DFO, também levamos em consideração as necessidades de usuários que podem ter orçamento restrito e que precisam tomar decisões baseados não apenas na precisão final do processo, como é comum em Otimização Contínua, mas também nas precisões intermediárias, em termos do orçamento disponível. Uma diferença é que, aqui, nosso custo central reside na SVD, enquanto que em DFO o gasto está concentrado nas avaliações da função objetivo. Além disso, por haver uma clara sensibilidade do número de avaliações de função à quantidade de variáveis de otimização, seus autores optaram por incorporar a dimensão do problema à unidade de custo, utilizando nos *data profiles* a relação entre a medida de desempenho e o número de variáveis de otimização. Para entender melhor o uso dos *profiles*, veja [70] ou, em português, [71, Cap. 5].

O uso de *data profiles* requer uma definição do significado de *resolver* um problema. Em geral, diz-se que um problema foi resolvido se determinado teste de convergência foi satisfeito, sendo que normalmente o teste é baseado em alguma medida de otimalidade. Como no nosso caso não é trivial estabelecer medidas de otimalidade compatíveis com ambos métodos, vamos instituir na Seção 5.3 o que chamaremos de solução de referência e, assim, será possível tratar diretamente de erros enquanto medidas de desempenho. Por ora, vamos assumir que dispomos de uma solução para cada problema-teste. Com isso, estabelecemos a Definição 10 de problema resolvido.

**Definição 10** (Problema Resolvido). Dizemos que um problema  $p$  foi resolvido por determinado método de otimização  $s$  com solução  $X$  quando seu erro relativo à solução de referência  $X^*$  for menor do que ou igual a certa tolerância  $\tau > 0$ , isto é, se

$$ER(X; X^*) \leq \tau. \quad (4.6)$$

Dizemos também que na resolução de  $p$  por meio de  $s$  houve um custo  $t_{p,s}$  em número de SVDs quando  $X$  for o primeiro iterando para o qual ER satisfaz o Teste de Convergência (4.6).

Note que a condição em (4.6) equivale a exigir que a distância do iterando à solução seja no máximo igual a uma fração do tamanho da solução, ambas na norma de Frobenius:

$$ER(X; X^*) \leq \tau \iff \|X - X^*\|_F \leq \tau \|X^*\|_F. \quad (4.7)$$

### Boxplots

Ao longo da apresentação dos resultados, lançaremos mão dos chamados diagramas de caixa (*boxplots*), os quais descrevem graficamente informações sobre o conjunto de dados em questão, sem suposições prévias a respeito da distribuição estatística subjacente. A título de ilustração, veja a Figura 3, que contém os diagramas de caixa das entradas matriciais dos problemas-testes aleatórios.

De maneira visual, um diagrama de caixa permite a estimativa de cinco grandezas que ajudam a descrever a distribuição dos dados: os quartis 0, 1, 2, 3 e 4. Nos *boxplots* que empregamos neste trabalho, há um retângulo em azul (caixa) que corresponde ao intervalo que se estende do quartil 1 ao 3 – demarcando portanto o intervalo interquartil com os 50% dos elementos centrais. Esse mesmo retângulo contém um segmento de reta vertical em vermelho que representa o quartil 2, ou seja, a mediana do conjunto.

A amplitude do intervalo interquartil (digamos,  $|I|$ ) é usada para classificar como *outliers* os pontos mais afastados da mediana, se houver, da seguinte maneira: é considerado *outlier* todo o ponto que estiver para além da margem de  $1.5 \cdot |I|$  acima do terceiro quartil ou abaixo do primeiro quartil. Nas figuras, os *outliers* são graficados por meio do símbolo “+”, em vermelho.

A linha tracejada, conhecida como *whisker* ou fio de bigode, se prolonga pela extensão dos dados encarados como condizentes com o padrão da distribuição, cujos extremos são denominados quartis 0 e 4, e, no caso de uma distribuição normal padrão, abrangem mais de 99% dos pontos.

## 4.2 Construção das Instâncias Aleatórias

Em estudos numéricos de métodos de otimização é comum a adoção de instâncias artificiais aleatórias especialmente construídas para viabilizar ou favorecer os testes computacionais, de acordo com o que se deseja investigar. No nosso caso, tal construção permite que tenhamos a garantia de lidar com instâncias que satisfazem as hipóteses de recuperabilidade e unicidade de soluções, além de deixar à nossa disposição, de antemão, as próprias soluções em si.

Como vimos na primeira parte deste trabalho, Chandrasekaran *et al.* ([25]) desenvolveram o conceito de incoerência posto-esparsidade, um princípio de incerteza determinístico entre o padrão de esparsidade de uma matriz e seus espaços linha e coluna, a fim de estabelecer condições suficientes que garantam a unicidade da solução do problema de decomposição matricial (2.5). Tais condições são enunciadas em termos de duas medidas relacionadas à matriz  $A$  e o que se demonstra é que há recuperação, para  $t$  em certo intervalo, quando o produto dessas quantidades é *pequeno*. A propósito, a publicação de Chandrasekaran *et al.* serviu como base para Yuan e Yang [103] realizarem seus experimentos numéricos e constituírem suas instâncias aleatórias.

Uma outra abordagem para o tema da unicidade, com condicionantes e consequências levemente diferentes, pode ser encontrada em [21, Seção 1]. Em termos gerais, o que se exige nesse trabalho é que a componente associada ao posto tenha vetores singulares *razoavelmente espalhados* no espaço e que a componente esparsa, entendida como ruído, tenha entradas não nulas cujas posições na matriz são aleatórias. Em outras palavras, nem a componente esparsa pode ter posto baixo e nem os vetores singulares da componente de posto baixo podem ser muito esparsos. Dessa forma, a recuperação das componentes matriciais a partir do problema de otimização (com um valor universal para  $t$ ) ocorre com alta probabilidade.

Apesar da semelhança entre as abordagens de [25] e de [21], nesta última é estabelecido um valor único do parâmetro  $t$  para o qual a recuperação fica assegurada com alta probabilidade, enquanto que, na primeira, o valor de  $t$  depende dos dados e pode ter que ser obtido por meio da resolução de problemas de otimização. O valor do parâmetro  $t$  em Candès *et al.*<sup>2</sup> no caso geral  $m \times n$  é (cf. [21, p. 11:6, 11:21]):

$$t = \frac{1}{1 + \sqrt{\max\{m, n\}}}. \quad (4.8)$$

Para problemas práticos, esse valor pode ainda ser ajustado empiricamente para prover melhores resultados.

Em nossos testes, optamos por adotar as construções de caráter aleatório presentes nos trabalhos de Yuan e Yang [103] e Candès *et al* [21], as quais, por simplicidade, geram apenas matrizes quadradas. Seguindo esses autores, primeiro são construídas as soluções  $S^*$  e  $L^*$  com as características de posto e esparsidade desejadas, para depois somá-las, obtendo-se  $A$ . Em ambos os casos, o posto de  $L^*$  é garantido através do produto de duas matrizes,  $L_1$  e  $L_2$ , de posto completo e dimensões apropriadas, enquanto a esparsidade de  $S^*$  é imposta sorteando-se a posição de uma quantidade predeterminada de entradas não nulas e atribuindo-lhes um valor aleatório de acordo com certa distribuição.

<sup>2</sup> Para  $t \in (0, 1)$  e  $\gamma \in (0, \infty)$  temos que  $\gamma = t/(1-t) \iff t = \gamma/(1+\gamma)$ .

Yuan e Yang geram dois tipos de instâncias aleatórias, classificadas em **impulsivas** ou **Gaussianas**, a depender da distribuição dos valores das entradas não nulas de  $S^*$ . Por outro lado, Candès *et al.* adotam uma forma de construção semelhante à das instâncias impulsivas de Yuan e Yang, embora a classificação não se faça necessária naquele trabalho porque há essencialmente um único tipo de instância aleatória considerado por seus autores.

Outra diferença entre os dois trabalhos mencionados está nas configurações de dimensão, esparsidade e posto contempladas em cada um deles. Enxergando a matriz  $S$  como um erro incidente na componente de posto baixo  $L$ , Candès *et al.* constroem instâncias de dimensão  $n \times n$ , com  $n$  variando no conjunto  $\{500, 1000, 2000, 3000\}$ , nas quais  $L^*$  tem posto percentual  $rkr$  fixado em 5%, isto é, o posto  $r$  é dado por  $r = 0.05n$ , ao passo que a taxa de esparsidade  $spr$  de  $S^*$  é alternada entre 5% e 10%. Já Yuan e Yang, por outro lado, além de diversificar a dimensão  $n$  no conjunto  $\{100, 500, 1000\}$  e a taxa de esparsidade  $spr$  também alternada entre 5% e 10%, variam o posto percentual  $rkr$  com duas taxas que podem valer entre 2% e 10% a depender da dimensão. Para padronizar as características de posto e esparsidade nessas instâncias, fixamos os postos percentuais  $rkr$  em 5% ou 10%.

Em nossos experimentos, a fim de possibilitar comparações, incorporamos os três tipos de construção matricial, além de todas as configurações de dimensão, posto e esparsidade presentes em algum dos dois trabalhos. No entanto, vale ressaltar que nossas matrizes não são exatamente as mesmas desses autores, já que são resultantes de processos (pseudo)aleatórios.

A Tabela 1 contém todas as combinações abordadas e alguns rótulos que eventualmente auxiliam na menção a instâncias específicas. Por praticidade, adotaremos os termos **CLMW**, **YYimp** e **YYgaus** para fazer referência, respectivamente, ao grupo de instâncias de Candès *et al.* e aos grupos impulsivo e Gaussiano de Yuan e Yang. Os detalhes das construções aleatórias, assim como os códigos em MATLAB que as geraram, são apresentados a continuação.

## Estruturação das matrizes aleatórias

Sejam  $n$ ,  $spr$  e  $rkr$ , respectivamente, a dimensão do problema, a taxa de esparsidade da componente esparsa e o posto percentual da componente de posto baixo. Sejam, ainda, as matrizes  $L_1, L_2^T \in \mathbb{R}^{n \times r}$  com entradas aleatórias independentes e identicamente distribuídas obtidas de uma distribuição normal  $\mathcal{N}(0, \sigma^2)$  com média zero, desvio padrão  $\sigma$  e, conseqüentemente, variância  $\sigma^2$ . Então, construímos a componente  $L^*$  da solução do problema fazendo

$$L^* = L_1 L_2. \quad (4.9)$$

Tabela 1 – Instâncias de Teste Aleatórias: correspondência rótulo-instância.

Rótulo		m	n	spr	rkr	rnk
YYimp-1	YYgaus-1	100		5	5	5
YYimp-2	YYgaus-2			10	10	10
YYimp-3	YYgaus-3			10	5	5
YYimp-4	YYgaus-4				10	10
YYimp-5	YYgaus-5	500		5	5	25
YYimp-6	YYgaus-6			10	10	50
YYimp-7	YYgaus-7			10	5	25
YYimp-8	YYgaus-8				10	10
YYimp-9	YYgaus-9	1000		5	5	50
YYimp-10	YYgaus-10			10	10	100
YYimp-11	YYgaus-11			10	5	50
YYimp-12	YYgaus-12				10	10
CLMW-1		500		5	5	25
CLMW-2				10	10	
CLMW-3		1000		5	5	50
CLMW-4				10	10	
CLMW-5		2000		5	5	100
CLMW-6				10	10	
CLMW-7		3000		5	5	150
CLMW-8				10	10	

Para a componente esparsa, primeiro determinamos as  $s = spr \cdot n^2$  posições das entradas de  $S^*$  que poderão assumir valores não nulos, ou seja, estabelecemos o conjunto  $\Omega$  de suporte de  $S^*$  (com cardinalidade  $s$ ), o que será feito de maneira uniformemente aleatória. Em seguida, resta atribuir valores às entradas, o que também é feito de maneira aleatória, de acordo com alguma distribuição de probabilidade.

Em termos matemáticos, podemos representar esse processo através do operador matricial  $\mathcal{P}_\Omega$  de projeção sobre o espaço das matrizes com suporte  $\Omega$ , cuja ação sobre uma matriz  $E$  com entradas aleatórias pode ser definida por

$$[\mathcal{P}_\Omega(E)]_{ij} = \begin{cases} E_{ij}, & \text{se } (i, j) \in \Omega \\ 0, & \text{se } (i, j) \notin \Omega \end{cases}. \quad (4.10)$$

Assim, sendo  $E$  uma matriz cujas entradas são obtidas de variáveis aleatórias independentes com distribuições idênticas, construímos a componente  $S^*$  como

$$S^* = \mathcal{P}_\Omega(E). \quad (4.11)$$

Isso é o que há de comum entre os três tipos de construção de instâncias aleatórias. A seguir, descrevemos as particularidades de cada um dos três grupos.

Construção segundo Candès *et al.* (CLMW)

A variância das distribuições normais usadas na construção de  $L_1$  e  $L_2$  é  $\sigma^2 = 1/n$ , isto é,

$$[L_1]_{ij}, [L_2]_{ij} \sim \mathcal{N}(0, 1/n). \quad (4.12)$$

A matriz  $E$  usada na construção de  $S^*$  admite entradas  $+1$  e  $-1$  de acordo com variáveis aleatórias independentes em experimentos de Bernoulli com probabilidade 0.5, ou seja,

$$[E]_{ij} \in \{-1, 1\} \quad (\text{resultados equiprováveis}) \text{ e} \quad (4.13)$$

$$S^* = \mathcal{P}_\Omega(E) \in \{0, \pm 1\}^{n \times n}.$$

Construção segundo as versões de Yuan e Yang. (YY)

As distribuições usadas na construção de  $L_1$  e  $L_2$  são normais padrão, isto é,

$$[L_1]_{ij}, [L_2]_{ij} \sim \mathcal{N}(0, 1). \quad (4.14)$$

- Impulsiva (YYimp):

A matriz  $E$  usada na construção de  $S^*$  admite entradas com valores  $+\|\text{vec}(L^*)\|_\infty$  e  $-\|\text{vec}(L^*)\|_\infty$  de acordo com variáveis aleatórias independentes em experimentos de Bernoulli com probabilidade 0.5, ou seja,

$$[E]_{ij} \in \{-\|\text{vec}(L^*)\|_\infty, \|\text{vec}(L^*)\|_\infty\} \quad (\text{resultados equiprováveis}) \text{ e} \quad (4.15)$$

$$S^* = \mathcal{P}_\Omega(E) \in \{0, \pm \|\text{vec}(L^*)\|_\infty\}^{n \times n}.$$

- Gaussiana (YYgaus):

A matriz  $E$  usada na construção de  $S^*$  admite entradas com valores obtidos de uma distribuição normal padrão (Gaussiana), ou seja,

$$[E]_{ij} \sim \mathcal{N}(0, 1) \quad \text{e} \quad S^* = \mathcal{P}_\Omega(E) \in \mathbb{R}^{n \times n}. \quad (4.16)$$

No Apêndice B, fornecemos os Algoritmos 5, 6 e 7, com os códigos das construções das instâncias aleatórias no MATLAB.

### 4.3 Validação da implementação do ADMM

O código do ADMM utilizado em nossos testes foi implementado de acordo com os algoritmos disponíveis nos trabalhos [21] e [103]. Com o objetivo de validar nossa implementação, testamos nossa versão do ADMM em problemas aleatórios gerados segundo os critérios desses autores. Para facilitar a comparação, reproduzimos nas Tabelas 37, 38 e 39, no Anexo A, os dados originalmente publicados.

A configuração do ADMM foi realizada com os parâmetros exibidos na Tabela 2. Dentre eles, convém destacar que o valor do parâmetro  $\rho$  de penalização será calculado da mesma maneira ao longo deste trabalho, conforme a fórmula sugerida em [103] e igualmente adotada em [21]. Com essa escolha,  $1/\rho$  equivale a quatro vezes a média dos valores absolutos das entradas de  $A$ . Quanto à terminação do ADMM, em cada caso seguiram-se os mesmos critérios das publicações originais. Em Yuang e Yang [103], a execução foi interrompida quando o tamanho relativo do passo atingiu um valor inferior a  $10^{-4}$ . Já em Candès *et. al* [21], a parada ocorreu quando a infactibilidade relativa foi menor do que ou igual a  $10^{-7}$ .

Os resultados dos testes com as instâncias impulsivas e Gaussianas análogas às de Yuan e Yang [103], assim como os resultados com as instâncias análogas às de Candès *et. al* [21], encontram-se nas Tabelas 3, 4 e 5.

Tabela 2 – Parâmetros de Validação do ADMM

$S^{(0)} = L^{(0)} = Z^{(0)} = 0_{n \times n}$	$\rho = \frac{n^2}{4 \ \text{vec}(A)\ _1}$	$M_{\text{it}} = 1000$	$\epsilon_{\text{relinf}} = 10^{-7}$ (CLMW)
			$\epsilon_x = 10^{-4}$ (YY)

Como esperado, os valores obtidos não são idênticos aos dos respectivos autores, já que não estamos lidando com as mesmas matrizes, mas sim com instâncias construídas de acordo com os mesmos critérios. Além disso, a diferença entre as máquinas em que aqueles testes foram realizados e a nossa torna improvável a reprodução dos valores publicados. No entanto, os resultados obtidos com nossa implementação são totalmente condizentes com o que foi reportado, no sentido da ordem de grandeza das medidas comparadas, principalmente das mais relevantes neste primeiro momento: a quantidade de iterações, os erros relativos, o posto e a esparsidade.

A qualidade dos resultados obtidos com o ADMM nas instâncias aleatórias é excelente; atingem-se grandes precisões com um custo muito baixo em termos da quantidade de decomposições em valores singulares. Impressiona ainda mais o fato de que o aumento da dimensão das instâncias não reflete em um acréscimo considerável na quantidade de SVDs gasta, em especial com as instâncias CLMW. Na seção a seguir, investigamos particularidades das estruturas das matrizes aleatórias de forma que possam justificar o fenômeno que acabamos de descrever. Além disso, caracterizamos tais instâncias para que possamos compará-las com as de videovigilância e, assim, fazer ponderações sobre a representatividade de instâncias aleatórias e, conseqüentemente, elaborar melhores estratégias de resolução de problemas práticos.

Tabela 3 – Instâncias aleatórias impulsivas baseadas em Candès *et al.* ([21]).

m	n	spr	rnk	t(%)	nIter	ObjV	ErrSP	ErrLR	RelErr	spr	≈spr	rnk	≈rnk	Duração (s)	CPU (s)	Parada
500	500	5	25	4	16	558.86562	8.6e-08	9.8e-07	9.6e-08	5	5	25	25	1.9e+00	2.5e+01	relinfeas
500	500	10	25	4	17	1093.74790	5.6e-08	1.0e-06	6.4e-08	10	10	25	25	2.2e+00	2.8e+01	relinfeas
1000	1000	5	50	3	15	1580.39156	1.1e-07	1.6e-06	1.2e-07	5	5	50	50	7.8e+00	9.0e+01	relinfeas
1000	1000	10	50	3	16	3113.13635	7.4e-08	1.8e-06	8.4e-08	10	10	50	50	8.3e+00	9.9e+01	relinfeas
2000	2000	5	100	2	15	4470.99680	7.8e-08	1.6e-06	8.6e-08	5	5	100	100	4.3e+01	5.2e+02	relinfeas
2000	2000	10	100	2	15	8845.17073	9.4e-08	3.3e-06	1.1e-07	10	10	100	100	4.2e+01	5.0e+02	relinfeas
3000	3000	5	150	2	15	8213.93576	6.2e-08	1.6e-06	6.9e-08	5	5	150	150	1.4e+02	1.8e+03	relinfeas
3000	3000	10	150	2	15	16282.44063	7.6e-08	3.2e-06	8.6e-08	10	10	150	150	1.4e+02	1.8e+03	relinfeas

Tabela 4 – Instâncias aleatórias impulsivas baseadas em Yuan e Yang ([103]).

m	n	spr	rnk	t(%)	nIter	ObjV	ErrSP	ErrLR	RelErr	spr	≈spr	rnk	≈rnk	Duração (s)	CPU (s)	Parada
100	100	5	5	9	21	930.72891	1.2e-04	8.7e-05	1.1e-04	5	5	5	5	3.4e-01	2.9e+00	relstep
100	100	5	10	9	21	1554.49001	8.8e-05	6.5e-05	7.8e-05	5	5	10	10	1.9e-01	2.1e+00	relstep
100	100	10	5	9	19	1443.52406	8.3e-05	9.2e-05	8.6e-05	10	10	5	5	2.6e-01	2.8e+00	relstep
100	100	10	10	9	23	2179.86400	1.8e-04	2.4e-04	2.0e-04	10	10	10	10	3.8e-01	4.4e+00	relstep
500	500	5	25	4	22	26870.18563	5.2e-05	3.4e-05	4.6e-05	5	5	25	25	3.8e+00	4.3e+01	relstep
500	500	5	50	4	26	43800.94757	7.0e-05	5.4e-05	6.4e-05	5	5	50	50	2.9e+00	3.6e+01	relstep
500	500	10	25	4	23	39124.11807	7.7e-05	7.5e-05	7.6e-05	10	10	25	25	5.0e+00	5.3e+01	relstep
500	500	10	50	4	27	60533.48263	8.7e-05	9.7e-05	9.0e-05	10	10	50	50	7.0e+00	7.4e+01	relstep
1000	1000	5	50	3	26	106072.05973	3.2e-05	1.9e-05	2.8e-05	5	5	50	50	4.7e+01	3.9e+02	relstep
1000	1000	5	100	3	31	173449.48309	5.9e-05	4.2e-05	5.2e-05	5	5	100	100	8.4e+01	6.0e+02	relstep
1000	1000	10	50	3	26	156551.07270	4.6e-05	4.2e-05	4.5e-05	10	10	50	50	5.8e+01	4.3e+02	relstep
1000	1000	10	100	3	31	252924.48378	9.6e-05	1.1e-04	9.9e-05	10	10	100	100	1.7e+01	1.8e+02	relstep

Tabela 5 – Instâncias aleatórias Gaussianas baseadas em Yuan e Yang ([103]).

m	n	spr	rnk	t(%)	nIter	ObjV	ErrSP	ErrLR	RelErr	spr	≈spr	rnk	≈rnk	Duração (s)	CPU (s)	Parada
100	100	5	5	9	26	471.46563	2.8e-03	1.1e-04	3.1e-04	5	5	5	5	7.2e-01	6.7e+00	relstep
100	100	5	10	9	35	919.92838	6.6e-03	2.3e-04	5.2e-04	5	5	12	12	9.9e-01	9.0e+00	relstep
100	100	10	5	9	31	557.65603	3.5e-03	1.7e-04	4.8e-04	10	10	8	8	6.9e-01	5.9e+00	relstep
100	100	10	10	9	33	997.82051	6.5e-03	3.6e-04	7.1e-04	10	10	13	13	9.1e-01	8.2e+00	relstep
500	500	5	25	4	32	12165.60318	5.7e-03	8.8e-05	2.7e-04	5	5	32	32	1.5e+01	1.4e+02	relstep
500	500	5	50	4	38	23662.99430	1.0e-02	1.5e-04	3.5e-04	5	5	65	65	2.0e+01	1.6e+02	relstep
500	500	10	25	4	35	12541.33471	5.1e-03	1.2e-04	3.4e-04	10	10	44	44	2.2e+01	1.5e+02	relstep
500	500	10	50	4	44	24329.62006	7.6e-03	1.8e-04	3.8e-04	11	11	80	80	2.7e+01	2.1e+02	relstep
1000	1000	5	50	3	35	48481.89314	7.4e-03	8.1e-05	2.5e-04	5	5	65	65	8.8e+01	6.4e+02	relstep
1000	1000	5	100	3	42	95460.59563	1.2e-02	1.3e-04	3.0e-04	5	5	132	132	6.2e+01	4.8e+02	relstep
1000	1000	10	50	3	38	50246.83467	6.2e-03	1.0e-04	3.0e-04	10	10	91	91	3.4e+01	3.0e+02	relstep
1000	1000	10	100	3	47	96163.20351	9.2e-03	1.5e-04	3.3e-04	11	11	162	162	2.7e+01	2.8e+02	relstep

## 4.4 Variabilidade nas Instâncias de Teste

O objetivo desta análise é levantar hipóteses a respeito da representatividade das instâncias aleatórias com relação ao desempenho do ADMM e do SPGM aplicados aos problemas de videovigilância. Sabemos da teoria de recuperabilidade que, para que seja possível obter a decomposição de uma matriz  $A$  em parcelas  $S^*$ , esparsa, e  $L^*$ , de posto baixo, é necessário que  $L^*$  não seja *muito* esparsa nem  $S^*$  tenha posto *muito* baixo. Tendo isso em mente, analisamos a seguir as distribuições das entradas e dos valores singulares das soluções  $S^*$  e  $L^*$  dos problemas aleatórios, assim como as de sua soma  $A$ .

Sob cada uma dessas perspectivas – das entradas e dos valores singulares –, estamos interessados em saber qual característica das duas componentes matriciais predomina na matriz resultante da adição. É de nosso interesse também interpretar essas predominâncias no contexto de transformação das matrizes em vídeos. Para tanto, analisamos especificamente as distribuições dos valores das entradas e dos valores singulares associadas às matrizes.

### Caracterização das Instâncias Aleatórias

Como vimos, a construção da matriz  $L$  de dimensão  $n \times n$  e posto  $r$  das instâncias aleatórias é feita através da multiplicação de duas matrizes de posto completo cujas entradas possuem distribuição normal com parâmetros predeterminados. Já a matriz esparsa  $S$ , de dimensão  $n \times n$  e esparsidade  $s$ , é construída a partir da seleção aleatória uniforme das posições que receberão como entradas valores não nulos.

Nas instâncias de *Candès et al.*, as entradas não nulas das matrizes esparsas foram sorteadas aleatoriamente dentre  $+1$  e  $-1$ . Note que essa construção se baseia na ocorrência de pulsos esparsos, assim como as instâncias impulsivas de Yuan e Yang. Nestas, porém, a magnitude das entradas não nulas de  $S$  é determinada pela maior entrada de  $L$  em módulo, sendo o sinal escolhido aleatoriamente. Por outro lado, nas instâncias gaussianas, as entradas não nulas da matriz  $S$  são obtidas a partir de uma distribuição normal padrão.

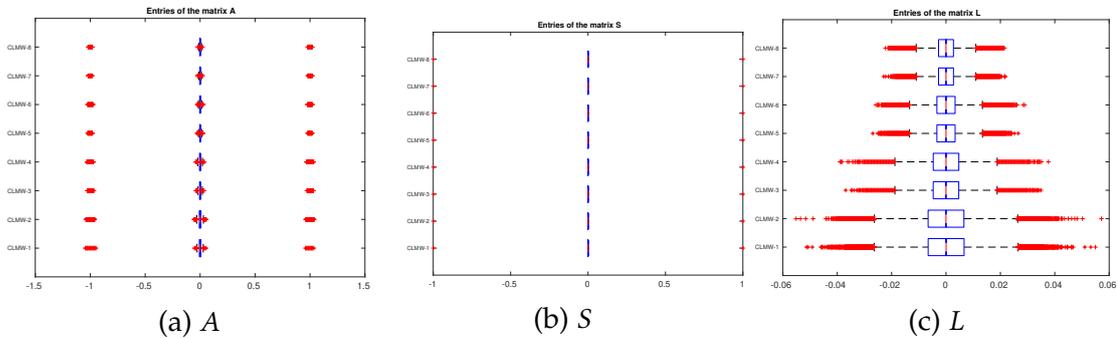
Por clareza e simplicidade, nos referimos às matrizes de *Candès et al.* como  $A^{\text{CLMW}}$ ,  $L^{\text{CLMW}}$  e  $S^{\text{CLMW}}$ ; às matrizes de Yuan e Yang, como  $A^{\text{YYimp}}$ ,  $S^{\text{YYimp}}$  e  $L^{\text{YYimp}}$ , no caso impulsivo, e  $A^{\text{YYgaus}}$ ,  $S^{\text{YYgaus}}$  e  $L^{\text{YYimp}}$ , no gaussiano. Quando a notação não diferenciar entre os autores e tipos é porque se refere a qualquer um deles.

### 4.4.1 Entradas Matriciais

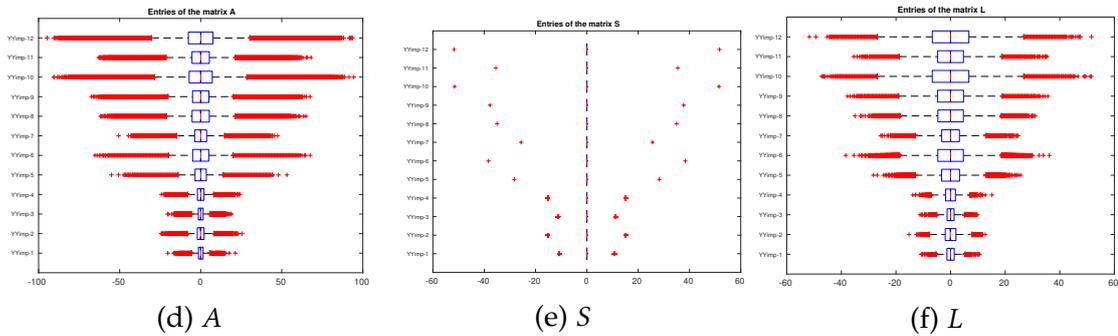
A Figura 3 ilustra a distribuição dos valores das entradas das matrizes  $A$ ,  $S$  e  $L$  para cada um dos três grupos de instâncias aleatórias. A constatação mais evidente sobre os gráficos, que não chega a ser surpreendente, é a tendência à simetria em torno do zero. Além disso, de um modo geral, vemos em cada tipo de construção matricial que a variação na dimensão do problema altera pouco ou nada o formato dos gráficos, e, portanto, nos atemos às diferenças decorrentes das três maneiras de gerar a matriz de entrada, e não das dimensões.

Figura 3 – *Boxplots* das Entradas das Matrizes Aleatórias. Nos cálculos estatísticos para a construção dos gráficos, todas as entradas são levadas em conta, inclusive as nulas.

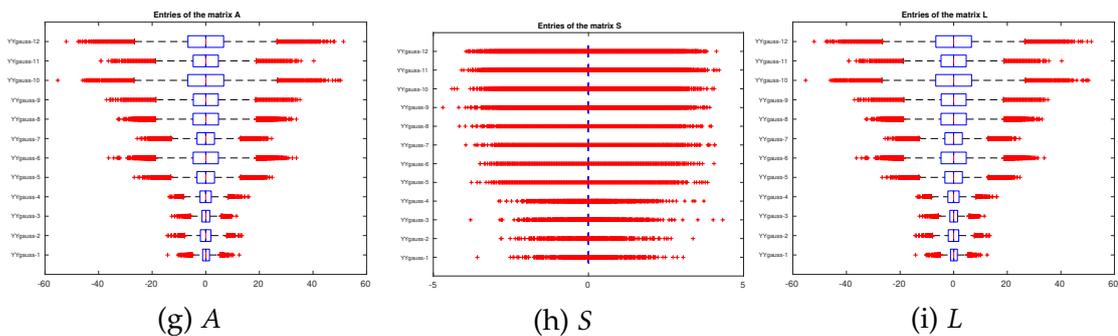
#### Instâncias Impulsivas de Candès, Li, Ma e Wright



#### Instâncias Aleatórias Impulsivas de Yuan e Yang



#### Instâncias Aleatórias Gaussianas de Yuan e Yang



Como a parcela  $S$  é esparsa por construção, poucos de seus elementos

se combinam efetivamente às entradas de  $L$  para gerar  $A$ . No entanto, os aspectos magnitude e sinal (positivo ou negativo) das entradas não nulas de  $S$ , combinados com os das entradas de  $L$ , também impactam na soma.

Podemos observar, nos gráficos da primeira fileira horizontal da figura, o efeito em  $A^{\text{CLMW}}$  da discrepância entre as ordens de grandeza das entradas das componentes matriciais: há uma nítida separação entre os efeitos dos elementos de  $S^{\text{CLMW}}$  e de  $L^{\text{CLMW}}$ . De fato, podemos interpretar  $A^{\text{CLMW}}$  como  $S^{\text{CLMW}}$  acrescida de uma pequena perturbação  $L^{\text{CLMW}}$ .

Por outro lado, à diferença de  $A^{\text{CLMW}}$ , mesmo também tendo caráter impulsivo, as matrizes  $S^{\text{Yimp}}$  da segunda fileira horizontal de gráficos apenas “esticam” levemente o formato das distribuições de  $L^{\text{Yimp}}$  para gerar  $A^{\text{Yimp}}$ . Isto ocorre porque as entradas não nulas de  $S^{\text{Yimp}}$  foram escolhidas de forma a replicar os maiores elementos de  $L^{\text{Yimp}}$  em módulo, e não a extrapolá-los, como ocorreu significativamente com as instâncias geradas conforme Candès *et al.* Portanto, a coincidência de sinais e magnitudes de mesma ordem de grandeza causaram esse efeito de esticamento.

Já com relação ao tipo gaussiano de construção, na terceira linha de gráficos, vemos que o efeito de  $S^{\text{Ygaus}}$ , cujas entradas são relativamente pequenas, é “diluído” na distribuição das entradas de  $A^{\text{Ygaus}}$ , como era de se esperar pela maneira como  $S^{\text{Ygaus}}$  foi estruturada. De fato, as amplitudes das entradas de  $S^{\text{Ygaus}}$  são razoavelmente comparáveis às larguras das caixas dos *boxplots* de  $L^{\text{Ygaus}}$ .

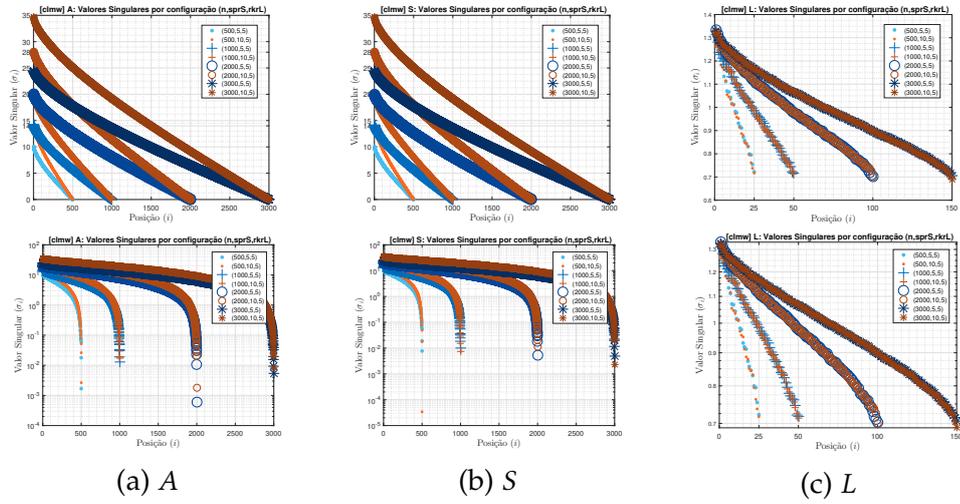
#### 4.4.2 Valores Singulares Matriciais

A Figura 4 exibe os valores singulares<sup>3</sup> de todas as instâncias de teste aleatórias. Os gráficos estão separados de acordo com os três grupos de estratégia de construção das matrizes, sendo que em cada grupo as colunas se referem às matrizes  $A$ ,  $S$  e  $L$ , no sentido da esquerda para a direita. Cada gráfico contém as marcações dos valores singulares das diferentes instâncias dentro do mesmo subgrupo, variando dimensão, esparsidade percentual e, se for o caso, posto percentual. Para facilitar a visualização dos dados, os valores singulares são apresentados tanto em escala linear (gráfico de cima) quanto logarítmica (gráfico de baixo). Destacamos que, mais do que prover cada valor singular, o objetivo da figura é fornecer valores aproximados e esboçar tendências de (des)continuidade. Nesse sentido, as tonalidades das cores complementam o tamanho e o formato dos marcadores para facilitar a visualização.

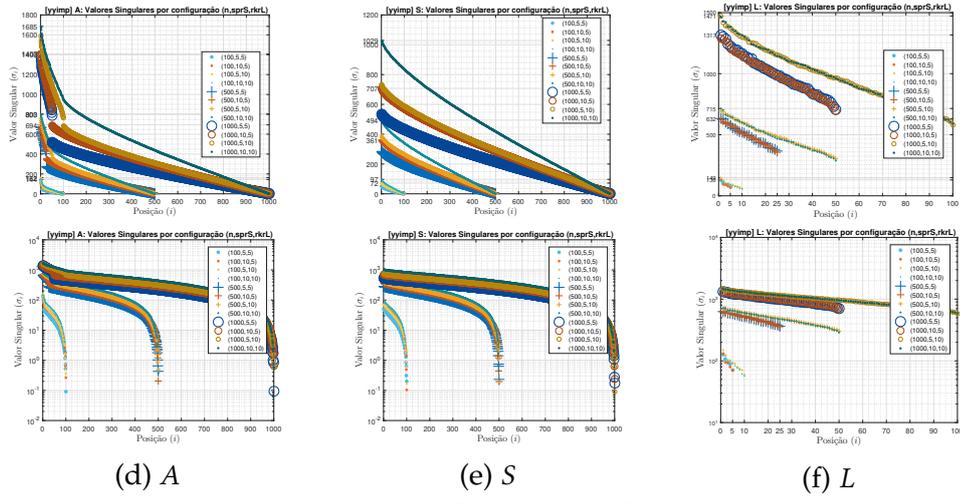
A primeira observação a respeito dos gráficos da Figura 4 é que, em todos os grupos, as marcações dos valores singulares de  $S$  e  $L$  formam um traçado com aspecto aproximadamente linear, ocupando uma determinada faixa de valores. Em

<sup>3</sup> Adotamos a convenção de que valores singulares  $\sigma_i$  são sempre números positivos, embora o vetor de valores singulares  $\sigma$  possa conter elementos nulos.

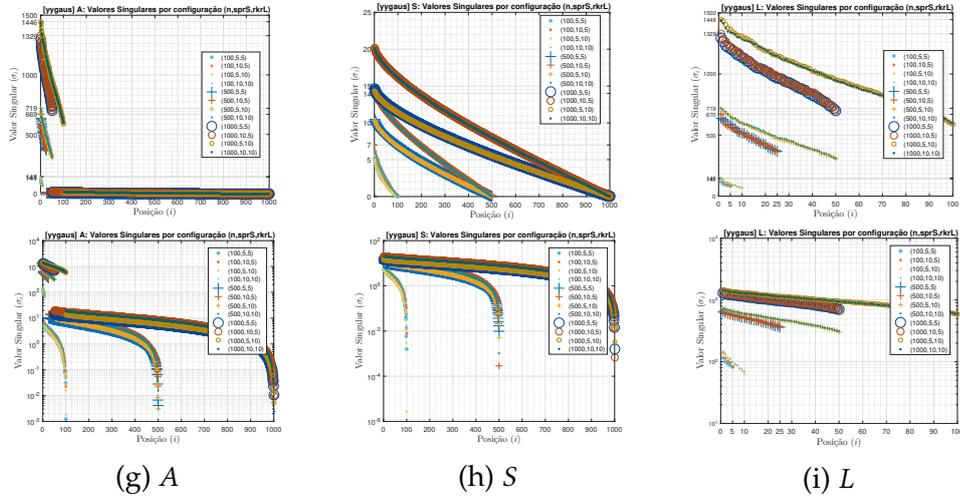
Figura 4 – Valores Singulares das Instâncias Aleatórias  
Instâncias Aleatórias de Candès, Li, Ma e Wright



Instâncias Aleatórias Impulsivas de Yuan e Yang



Instâncias Aleatórias Gaussianas de Yuan e Yang



particular, observando a terceira coluna, vemos que os valores singulares de  $L^{\text{CLMW}}$ , independentemente de dimensão ou de taxa de esparsidade, preenchem a faixa de 0.7 a cerca de 1.4. A diferença com  $L^{\text{Yimp}}$  e  $L^{\text{Ygaus}}$  é que, nestes casos, as faixas aumentam os valores de seus extremos com a dimensão do problema.

Apesar dessas semelhanças, é importante notar as mudanças de escala de um gráfico para outro. Enquanto os valores singulares de  $L^{\text{CLMW}}$  sequer chegam ao número 1.5, os de  $L^{\text{Yimp}}$  e  $L^{\text{Ygaus}}$  ora se mantêm na casa das centenas, ora chegam até na casa do milhar. Isso, balanceado com a ordem de grandeza dos valores singulares das respectivas componentes esparsas, impacta o aspecto da distribuição dos valores singulares das matrizes  $A$ .

De um modo geral, em um extremo, os valores singulares de  $A^{\text{CLMW}}$  são dominados pelos de  $S^{\text{CLMW}}$ , enquanto os de  $A^{\text{Ygaus}}$ , no extremo oposto, são regidos principalmente pela influência de  $L^{\text{Ygaus}}$ . No meio termo, temos as instâncias impulsivas de Yuan e Yang, que contrabalançam as contribuições de cada parcela aos valores singulares de  $A^{\text{Yimp}}$ , justamente porque os de ambas têm ordens de grandeza semelhantes. Analisando numericamente, podemos constatar que, se definimos o vetor  $\sigma(A) \in \mathbb{R}^n$  como aquele formado pelos valores singulares ordenados e completado com zeros, então temos que

$$\sigma(A) \approx \max(\sigma(S), \sigma(L)) \quad (4.17)$$

para cada caso, sobretudo nas instâncias de *Candès et al.* e nas gaussianas de Yuan e Yang. Consideramos, também, outras aproximações para  $\sigma(A)$  individualmente:  $\sigma(S)$ ,  $\sigma(L)$ ,  $\sigma(S) + \sigma(L)$ ,  $(\sigma(S) + \sigma(L))/2$  e algumas variações dessas. Para todas as instâncias, a aproximação em (4.17) resultou nos menores erros relativos (na norma euclidiana vetorial).

Por fim, é curioso perceber que os valores singulares das instâncias gaussianas de Yuan e Yang apresentam um salto considerável próximo à posição que marca o posto da componente  $L^{\text{Ygaus}}$ . De fato, embora seja difícil determinar pelo gráfico, constatamos numericamente que o salto de magnitude na sequência de valores singulares ocorre exatamente da posição cujo número é o posto de  $L^{\text{Ygaus}}$  para a seguinte. Em situações semelhantes a essas, seria possível elaborar bons palpites para o posto da solução apenas com tais informações. Essa constatação será relevante na discussão de uma estratégia para escolha do ponto inicial, no Capítulo 8.

#### 4.4.3 Relação Entradas e Valores Singulares

Em resumo, o que pudemos constatar a respeito das distribuições das entradas e dos valores singulares das instâncias aleatórias é que se tratam de três configurações que diferem em qual componente matricial prepondera sobre a matriz

A em cada quesito.

As instâncias de *Candès et al.* foram estruturadas de forma que  $A^{\text{CLMW}}$  seja dominada por  $S^{\text{CLMW}}$  no que diz respeito a suas entradas e também a seus valores singulares. Em ambos aspectos, portanto, a matriz  $L^{\text{CLMW}}$  pode ser vista como uma leve perturbação imposta a  $S^{\text{CLMW}}$  que culminou em  $A^{\text{CLMW}}$ . No caso das instâncias de Yuan e Yang, outros fenômenos são observados. No tipo gaussiano, em que as entradas da parcela que já é esparsa são ainda escolhidas para serem pequenas, extraídas de uma distribuição normal padrão, o que domina a matriz  $A^{\text{YYgaus}}$  é a componente  $L^{\text{YYgaus}}$ , seja na distribuição das entradas ou nos maiores valores singulares. A influência dos valores singulares de  $S^{\text{YYgaus}}$  sobre  $A^{\text{YYgaus}}$  não é desprezível, porém a grande diferença em módulo entre os valores singulares de  $S^{\text{YYgaus}}$  e  $L^{\text{YYgaus}}$  a ameniza. Por sua vez, o tipo impulsivo representa um meio termo: a parcela com atributos modestos em magnitude –  $S^{\text{YYimp}}$ , no caso das entradas, e  $L^{\text{YYimp}}$ , no caso dos valores singulares – embora não modifique consideravelmente a distribuição dos respectivos atributos em  $A^{\text{YYimp}}$ , ao menos a afetam para, em certo sentido, “esticá-la”, “impulsioná-la”.

Por fim, como esperado, constatamos que em nenhum dos tipos de construção as matrizes  $L$  são esparsas nem as  $S$  têm posto baixo – pelo contrário, estas se mostraram, respectivamente, densas e de posto completo. Cálculos exatos e aproximados de posto e esparsidade das matrizes aleatórias foram disponibilizados para consulta nas Tabelas 21 a 26 do Apêndice C.

### Considerações sobre as estruturas aleatórias

Nesta seção, vimos que cada tipo de construção aleatória replica estruturas similares quando apenas a dimensão do problema é variada. Por isso, a adoção de somente um dos três tipos certamente traz um viés para os experimentos computacionais. Em particular, o grupo de instâncias CLMW é o único dos três em que a matriz  $A$  de dados se assemelha tanto a uma única das componentes em ambos aspectos analisados (entradas e valores singulares), ao ponto de ser possível obter importantes informações sobre a solução sem necessidade de métodos de otimização.

No outro extremo, em menor intensidade, a estrutura do grupo YYgaus permite a descoberta do posto ótimo de  $L^*$  também sem o uso de métodos de otimização. Ainda assim, sua adoção em conjunto com o grupo YYimp, de características intermediárias, parece ser uma boa estratégia de contemplar os tipos impulsivo e gaussiano.

Com essa visão estrutural dos problemas aleatórios que agora temos em mente, é oportuno questionar quão boas são as estratégias de aproximação de posto fixo via SVD para resolver o problema SLRMD, já que não acarretam em novos custos

de decomposição matricial, como adiantamos na Seção 2.8. Veremos a seguir que, com essa estratégia, é possível obter erros relativos da ordem de 1%, a depender do tipo de instância. Nos capítulos seguintes, argumentaremos que essa precisão não é suficiente para obter bons resultados com a aplicação de videovigilância. Ainda assim, no Capítulo 8, constataremos que essa técnica pode ter sua utilidade caso queiramos elaborar um procedimento heurístico de construção do ponto inicial.

## 4.5 Aproximações de posto fixo como solução das aleatórias

Um dos primeiros questionamentos que surgem sobre a resolução do nosso problema de decomposição diz respeito às aproximações de  $A$  advindas de sua SVD, reconhecidas por serem as melhores aproximações em determinadas normas quando o posto é fixado *a priori* (veja a Seção 2.8). Poderia ser o caso de que uma dessas aproximações de posto fixo para  $A$  fosse suficientemente boa para aproximar a componente-solução  $L^*$ , ou ainda a  $S^*$ , e, assim, conduzir a soluções para o problema sem que sequer fosse necessário realizar procedimentos de otimização matemática. Pensando na nossa unidade de custo, esse procedimento seria promissor por envolver apenas uma única decomposição do tipo SVD, realizada em  $A$ . Reforçamos, no entanto, que uma de nossas hipóteses é que não dispomos de nenhuma informação sobre o valor do posto de  $L^*$ . Ainda assim, analisamos a seguir a qualidade das aproximações de posto pré-fixado na resolução do problema de decomposição posto-esparcidade.

Em termos matemáticos, a estratégia das melhores aproximações (cf. Def. 6) consiste em considerar um dos cenários

$$L = A_k \quad \text{ou} \quad S = A_k$$

como aproximações para a solução  $(S^*, L^*)$ . Para satisfazer a restrição de igualdade, a escolha de aproximar  $L$  por  $A_k$  impõe que  $S$  assumo os valores de  $A - A_k$ . Da mesma forma, a opção de aproximar  $S$  por  $A_k$  torna obrigatório que  $L$  valha  $A - A_k$ . Como temos as soluções dos problemas aleatórios de antemão, podemos avaliar a qualidade de cada um dos dois “palpites fundamentados” calculando o erro cometido por componente. Dessa forma, escolhido o posto  $k$ , podemos calcular os erros relativos definidos em (4.1) da maneira descrita a seguir.

Se  $(S, L) = (A_k, A - A_k)$ , então

$$\text{ER}(S; S^*) = \frac{\|A_k - S^*\|_F}{\|S^*\|_F} \quad \text{e} \quad \text{ER}(L; L^*) = \frac{\|A - A_k - L^*\|_F}{\|L^*\|_F} = \frac{\|A_k - S^*\|_F}{\|L^*\|_F}. \quad (4.18)$$

No outro cenário, se  $(S, L) = (A - A_k, A_k)$ , então

$$\text{ER}(S; S^*) = \frac{\|A - A_k - S^*\|_F}{\|S^*\|_F} = \frac{\|A_k - L^*\|_F}{\|S^*\|_F} \quad \text{e} \quad \text{ER}(L; L^*) = \frac{\|A_k - L^*\|_F}{\|L^*\|_F}. \quad (4.19)$$

A Figura 5 exibe os erros relativos por componente, consequentes da estratégia de resolução via melhor aproximação  $A_k$  para  $A$ , no contexto das instâncias aleatórias de dimensão  $n = 1000$ , agrupadas por tipo de construção (CLMW, YYimp e YYgaus). Os eixos horizontais representam o posto  $k$  de  $A_k$  e variam nos inteiros de 1 a 1000, sendo que  $k = 1000 = n$  significa que  $A_k = A$  e, portanto, resta a matriz nula para a componente matricial complementar. As Subfiguras 5(a) e 5(b), da linha superior, correspondem à aproximação  $S = A_k$ , enquanto as Subfiguras 5(c) e 5(d), da linha inferior, correspondem à aproximação  $L = A_k$ . Por outro lado, as Subfiguras 5(a) e 5(c) se referem aos erros em  $S$  relativos à solução  $S^*$ , ao passo que as Subfiguras 5(b) e 5(d) ilustram os erros em  $L$  relativos à solução  $L^*$ . Os gráficos para as outras dimensões são muito semelhantes na forma, guardadas as diferenças dimensionais, e por concisão não serão exibidos.

Observe que as semelhanças entre os gráficos de uma mesma linha se justificam pelo fato de que a única diferença entre os valores que retratam é um fator constante para cada instância. De fato, basta ver a “simetria cruzada” dos numeradores nas expressões para os erros, nas Equações (4.18) e (4.19): entre  $ER(A_k, S^*)$  e  $ER(A_k, L^*)$ , e entre  $ER(A - A_k, S^*)$  e  $ER(A - A_k, L^*)$ . Portanto, sem perda de generalidade, podemos direcionar as análises para o Erro Relativo em  $L$ .

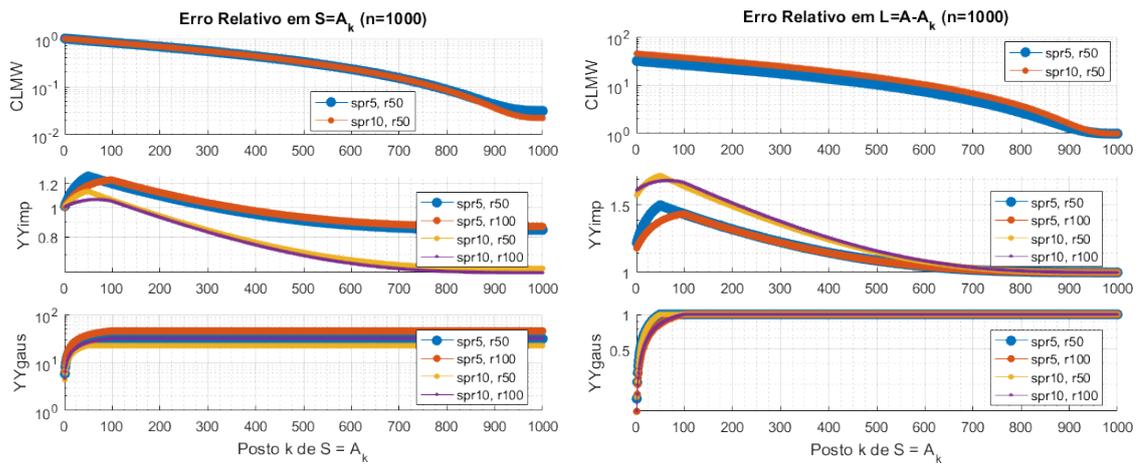
Assim, comparando os gráficos de cada um dos três grupos nas Subfiguras 5(b) e 5(d), vemos que cada curva na figura de cima apresenta tendência de variação (crescimento ou decrescimento) oposta à da sua curva correspondente na figura de baixo; uma espécie de complementaridade. Isso ocorre porque o que diferencia as curvas de cima das de baixo é a opção por qual das componentes  $S$  e  $L$  é aproximada por  $A_k$ . Com isso, a depender da similitude entre as matrizes  $A$  e  $S$ , e entre  $A$  e  $L$ , com relação a entradas e valores singulares, a aproximação  $A_k$  pode beneficiar uma opção em detrimento da outra, causando esse efeito de complementaridade na variação do erro.

Note que, se olharmos para o menor erro atingível em cada instância, podemos perceber pelos gráficos dos erros em  $L$  que a melhor estratégia para o grupo CLMW é aproximar  $S$  por  $A_k$ , com  $k$  próximo a 1000, isto é,  $S \approx A_{1000} = A$ , o que conduz a um erro relativo aproximadamente igual a 1 (veja na Subfigura 5(b)). Já para os grupos YYimp e YYgaus, ocorre o contrário: o melhor é aproximar  $L$  por  $A_k$ , neste caso com  $k$  próximo ao posto de  $L^*$ , e erros respectivamente da ordem de  $10^{-1}$  e  $10^{-2}$  (veja na Subfigura 5(d)). Uma diferença importante entre os grupos YYimp e YYgaus aqui é que, neste último, o erro se mantém mais ou menos estável para valores de  $k$  maiores do que os do posto de  $L^*$ , enquanto para o primeiro grupo os melhores erros só são atingidos para uma faixa estreita de valores de  $k$  em torno do posto ótimo.

É possível relacionar as constatações anteriores com a estrutura de cons-

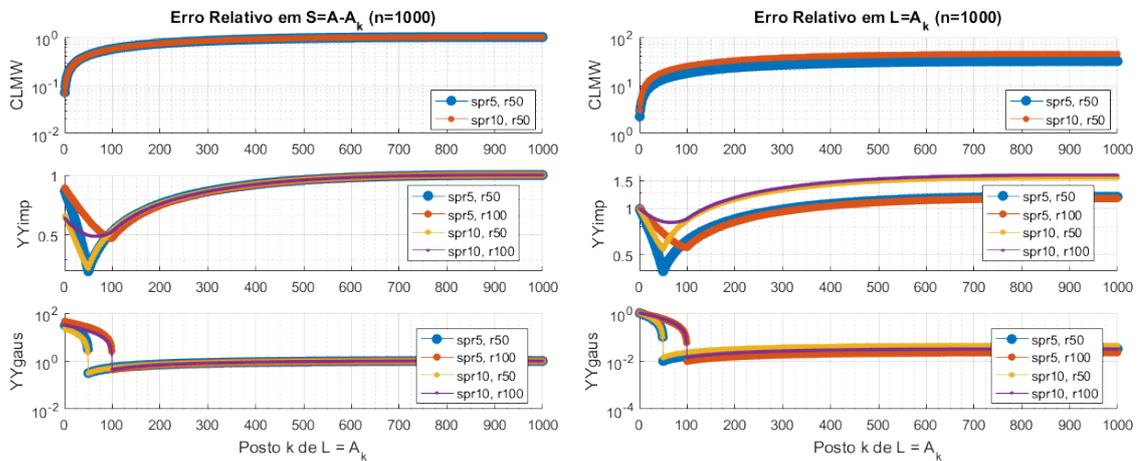
Figura 5 – Gráficos dos Erros Relativos em cada componente quando aproximamos uma delas por  $A_k$  – a melhor aproximação de posto  $k$  para  $A$  segundo a norma nuclear – e sua complementar por  $A - A_k$ , em termos do posto  $k$ , nos problemas aleatórios de dimensão  $n = 1000$ . A linha superior de subfiguras corresponde à aproximação  $S = A_k$  e a inferior, à  $L = A_k$ . A coluna de subfiguras à esquerda se refere aos erros em  $S$  relativos à solução  $S^*$ , enquanto a coluna da direita ilustra os erros em  $L$  relativos à solução  $L^*$ . Em cada subfigura, as curvas de erro estão agrupadas por tipo de construção (CLMW, YYimp ou YYgaus) e os eixos verticais estão em escala logarítmica.

Aproximações advindas da SVD de  $A$



(a)  $ER(A_k; S^*)$

(b)  $ER(A - A_k; L^*)$



(c)  $ER(A - A_k; S^*)$

(d)  $ER(A_k; L^*)$

trução das instâncias aleatórias, destrinchadas na Seção 4.4. Já havíamos identificado que, no grupo CLMW, a matriz  $A$  pode ser vista como uma pequena perturbação  $L^*$  aplicada à matriz esparsa  $S^*$ , o que justificaria a aproximação  $S = A_k$ . Por sua vez, nos grupos YYimp e YYgaus, as entradas e os maiores valores singulares de  $A$  são dominados pelos de  $L^*$  de formas um pouco distintas. As entradas matriciais de  $S^*$ , ainda que não sejam dominantes na soma, afetam mais as entradas de  $A$  no caso impulsivo do que no Gaussiano. Por outro lado, os maiores valores singulares de  $A$

são influenciados principalmente pelos de  $L^*$  em ambos os casos, ainda que no grupo YYgaus os valores singulares de  $L^*$  sejam consideravelmente maiores do que os de  $S^*$  – ao passo que, no grupo YYimp, a ordem de grandeza dos valores singulares de  $S^*$  é parecida com a de  $L^*$ . Isso não apenas reforça a similitude de  $A$  com  $L^*$  nesses dois grupos, corroborando a aproximação  $L = A_k$ , mas também atribui aos valores singulares uma das razões da dificuldade em obter o erro mínimo no grupo YYimp caso não se conheça o posto da solução  $L^*$ .

Esses resultados, junto com a caracterização das instâncias aleatórias, nos permitem constatar que os principais fatores a serem considerados na escolha da aproximação de uma das componentes matriciais por  $A_k$  são: a similitude entre  $A$  e as soluções-componentes nos quesitos entradas e valores singulares, e o posto da solução  $L^*$ .

Com tudo isso, a principal constatação a partir dessa análise é que nem nos casos mais promissores a estratégia da melhor aproximação de  $A$  resultou em erros relativos abaixo de 0.01, considerando o arredondamento até a segunda casa decimal, nem com relação a  $L$  e menos ainda com relação a  $S$ . Além disso, fica evidente que, mesmo na remota possibilidade de se conhecer o posto da solução  $L^*$  *a priori*, o valor de  $k$  que conduz aos menores erros depende da estrutura da solução de cada instância, como no exemplo do grupo CLMW.

Por último, paralelamente ao erro, podemos julgar também os aspectos posto e taxa de esparsidade, já que estão na raiz do nosso problema de decomposição. Se bem conseguimos garantir o posto de uma das componentes ao adotar para ela a aproximação  $A_k$ , sua esparsidade e a de seu complemento ficam completamente comprometidas. Para ilustrar, consideramos o caso em que  $L = A_k$  e calculamos numericamente as taxas *spr* de  $S = A - A_k$  para as instâncias com  $n = 1000$  e, mesmo considerando nulas as entradas com valor absoluto menor do que  $10^{-3}$ , os valores da taxa de esparsidade ultrapassaram os 93% em todas as instâncias, independentemente do número de valores singulares incorporados. Isso é muito além dos 5% ou 10% ideais para os casos em questão.

De todos os casos contemplados pelo que acabamos de discutir, um merece destaque especial por ser majoritariamente adotado neste e nos demais trabalhos como ponto inicial da execução dos métodos, que é o caso em que optamos pelas aproximações factíveis triviais:

$$(S, L) = (0, A), \quad (4.20)$$

$$(S, L) = (A, 0). \quad (4.21)$$

Note que, se  $(S, L) = (0, A)$ , então

$$\text{ER}(S; S^*) = 1 \quad \text{e} \quad \text{ER}(L; L^*) = \frac{\|A - L^*\|_F}{\|L^*\|_F} = \frac{\|S^*\|_F}{\|L^*\|_F}. \quad (4.22)$$

Analogamente, se  $(S, L) = (A, 0)$ , então

$$\text{ER}(S; S^*) = \frac{\|L^*\|_F}{\|S^*\|_F} \quad \text{e} \quad \text{ER}(L; L^*) = 1. \quad (4.23)$$

A Tabela 6 contém os valores dos erros referentes às instâncias aleatórias de dimensão  $n = 1000$ . Quando  $S = 0$  e, conseqüentemente,  $L = A$ , o erro relativo à solução-componente  $L^*$  é da ordem de  $10^1$ ,  $10^0$  e  $10^{-2}$  nos grupos CLMW, YYimp e YYgaus, respectivamente, nas instâncias aleatórias de teste deste trabalho. Por outro lado, quando  $L = 0$  e, portanto,  $S = A$ , o erro relativo à  $S^*$  é da ordem de  $10^{-2}$ ,  $10^{-1}$  e  $10^1$  nas mesmas instâncias dos grupos CLMW, YYimp e YYgaus, nessa ordem.

Tabela 6 – Erros Relativos nos casos em que cada uma das componentes-solução,  $S$  e  $L$ , é aproximada pela matriz nula (solução trivial), preservando a factibilidade por meio da componente complementar, nos problemas aleatórios de dimensão  $n = 1000$ . No primeiro caso,  $S$  é aproximada por 0 (logo, a aproximação  $\hat{L}$  precisa ser igual a  $A$ ) e os erros relativos cometidos com essa aproximação são iguais a 1, com relação a  $S$ , e  $\text{ER}(\hat{L}, L)$ , com relação a  $L$ . Analogamente, no segundo caso, quando  $L$  é aproximada por 0, forçando então  $\hat{S} = A$ , os erros relativos cometidos são iguais a  $\text{ER}(\hat{S}, S)$ , com relação a  $S$ , e 1, com relação a  $L$ .

Instância	spr	rnk	$\hat{L} = A$	$\hat{S} = A$
			$\text{ER}(\hat{L}; L)$	$\text{ER}(\hat{S}; S)$
CLMW-3	5	50	3.2e+1	3.2e-2
CLMW-4	10	50	4.5e+1	2.2e-2
YYimp-9	5	50	1.2e+0	8.5e-1
YYimp-10	5	100	1.2e+0	8.7e-1
YYimp-11	10	50	1.6e+0	6.3e-1
YYimp-12	10	100	1.6e+0	6.1e-1
YYgaus-9	5	50	3.2e-2	3.1e+1
YYgaus-10	5	100	2.2e-2	4.5e+1
YYgaus-11	10	50	4.5e-2	2.2e+1
YYgaus-12	10	100	3.2e-2	3.1e+1

O que acabamos de ver é que erros relativos da ordem de  $10^{-2}$  ou  $10^{-1}$  podem ser obtidos até mesmo por meio das soluções triviais, que podem ser vistas como um caso particular e simples da estratégia de melhores aproximações de  $A$ . Portanto, o uso sistemático dessa estratégia para resolver o problema de decomposição posto-esparsidade não encontra respaldo algum em nenhum dos tipos de instância aleatória considerados por nossas principais referências teóricas que estudam a recuperabilidade. Além disso, no geral, as soluções obtidas dessa forma não apresentam as características de posto e esparsidade almejadas, nem mesmo com o uso de tolerâncias no tamanho das entradas e dos valores singulares numericamente classificados como não nulos.

No Capítulo 8, veremos algumas possibilidades de investigação e aplicação das melhores aproximações de  $A$  em questões abertas sobre a construção de um ponto inicial que sirva como *warm-start* para o problema de decomposição e que, ao mesmo tempo, não dependa completamente da natureza do problema.

## 4.6 Experimentos Preliminares de Calibração com o SPGM

Na Seção 3.3, apresentamos a estrutura algorítmica geral do método que chamamos de SPGM. Vimos que cada ciclo homotópico, também chamado de iteração homotópica, corresponde ao conjunto de iterações ordinárias em que o parâmetro de suavização é mantido fixado. Para que possamos aplicar o SPGM, precisamos estabelecer os valores dos parâmetros dos quais seu funcionamento depende, em particular aqueles relacionados com a continuação homotópica. Nesta seção, discutimos e testamos algumas possibilidades para esses valores e para os critérios envolvidos, indicando alguns caminhos de investigação neste tema, que é intrinsecamente extenso e desafiador.

### Parâmetros

Podemos separar os parâmetros do SPGM essencialmente em três grupos: o da homotopia, o da busca linear e o dos testes de parada. Apesar de serem fundamentais para o desempenho do método, nosso foco neste trabalho não está na calibração desses parâmetros. No caso dos parâmetros de busca linear, nos restringimos a adotar valores baseados na vasta literatura disponível. Já na situação de escassez de dados publicados, como é o caso da homotopia, nos apoiamos em experimentos preliminares que nos ajudam a apontar direções a serem seguidas no uso dessa estratégia. Por sua vez, os parâmetros dos testes de parada são de escolha pragmática, como os de orçamento, ou são herdados da transição entre iterações homotópicas, como veremos a continuação.

#### Grupo 1: Parâmetros de Homotopia

Os parâmetros da homotopia são aqueles que definem a sequência de problemas parcialmente suavizados que serão atacados como sendo aproximações sucessivas e progressivas para o problema original. Podemos incluir nesse grupo os parâmetros que estabelecem a transição de uma iteração homotópica para outra, isto é, que indicam que o subproblema parcialmente suavizado foi suficientemente esgotado. São definidos a seguir.

- Parâmetro de suavização ( $\mu$ ) e quantidade de iterações homotópicas ( $T$ )

Seja  $(\mu_i)_{i=1}^T = (\mu_1, \mu_2, \dots, \mu_T)$  a sequência de valores para o parâmetro de suavização em cada iteração homotópica. Como queremos que a cada iteração homotópica o problema parcialmente suavizado se aproxime mais do problema original, essa sequência precisa ser decrescente. No entanto, há inúmeras possibilidades para as escolhas de cada um dos termos.

Para simplificar a análise, assumimos que cada termo  $\mu_{j+1}$  da sequência  $(\mu_i)_{i=1}^T$  pode ser obtido a partir de seu antecessor por meio da multiplicação de  $\mu_j$  por uma constante  $\eta \in (0, 1)$ , isto é,

$$\mu_{j+1} = \eta \mu_j, \quad j = 1, 2, \dots, T-1, \quad (4.24)$$

constituindo, assim, uma progressão geométrica decrescente. Podemos fazer com que toda a sequência dependa univocamente da tripla de parâmetros  $T$ ,  $\mu_1$  e  $\mu_T$ , já que seus termos podem ser expressos como

$$\mu_{j+1} = \eta^j \mu_1, \quad j = 1, 2, \dots, T-1,$$

em que

$$\eta = \sqrt[T-1]{\frac{\mu_T}{\mu_1}}. \quad (4.25)$$

Veremos que descrever a sequência por meio dessa tripla de parâmetros é conveniente pois cada um terá seu papel de adaptação ao tipo de instância a ser resolvida.

- Tolerância no tamanho relativo do passo no domínio ( $\epsilon_x$ )

Outro aspecto importante da homotopia é o critério de troca de uma iteração homotópica para outra, o qual exerce o papel de teste de parada do subproblema corrente. Como o SPGM resolve uma sequência de tais subproblemas, cada um com uma função objetivo distinta, optamos por um teste de parada baseado no tamanho (relativo) do passo no domínio, analogamente ao que fizeram Yuan e Yang em [103] sem homotopia:

$$\frac{\|X^{(k+1)} - X^{(k)}\|_F}{1 + \|X^{(k)}\|_F} < \epsilon_x. \quad (4.26)$$

Assim, a decisão sobre a troca da iteração fica condicionada ao valor do parâmetro  $\epsilon_x$ . Mesmo sendo um critério relativizado, não sabemos *a priori* quão bem cada subproblema homotópico precisa ser resolvido para que não se comprometa o desempenho do método como um todo.

### Grupo 2: Parâmetros de Busca Linear

Vimos no capítulo anterior que a busca linear ocorre quando o tamanho  $\alpha_+$  do passo espectral tentativo no sentido do deslocamento construído pelo SPGM conduz a um ponto  $X_+$  cujo valor objetivo  $\Psi_\mu(X_+)$  falha no critério não monotônico de decréscimo suficiente (cf. (3.19)), o qual a cada iteração  $k$  é validado pela desigualdade

$$\Psi_\mu\left(X_+^{(k+1)}\right) \leq \left( \max_{i=\max(k-M,0), \dots, k} \Psi_\mu\left(X^{(i)}\right) \right) - \frac{\sigma}{2} \alpha_+ \left\| X_+^{(k+1)} - X^{(k)} \right\|_F^2, \quad (4.27)$$

em que  $M + 1$  é a quantidade de iterações na janela de não monotonicidade e  $\sigma \in (0, 1)$  é o fator constante que ajuda a controlar o valor do decréscimo exigido. Há também um fator constante  $\zeta \in (0, 1)$  responsável pela diminuição do tamanho do passo durante a busca linear,  $\alpha_+ \leftarrow \alpha_+ \zeta$ , mas esse será simplesmente fixado em  $\zeta = 0.5$  ao longo deste trabalho. Além desses, há os parâmetros de salvaguarda  $\alpha_{max}$  e  $\alpha_{min}$  que garantem que  $\alpha$  se mantenha no intervalo  $(\alpha_{min}, \alpha_{max})$ . Aqui, adotaremos sempre  $\alpha_{min} = 10^{-30}$  e  $\alpha_{max} = 10^{30}$ .

### Grupo 3: Parâmetros dos Testes de Parada

O principal teste de parada do SPGM, indiretamente ligado à otimalidade, é a tolerância no tamanho do passo do domínio. Como esse critério é o mesmo da transição entre iterações homotópicas, podemos enxergá-lo como sendo um critério só, nos dois casos. Assim, o SPGM é interrompido quando o teste de parada da última iteração homotópica é satisfeito.

Outros parâmetros secundários relacionados a questões de interrupção de execução são os que representam limite no orçamento e que podem se referir ao número de iterações, à quantidade de decomposições do tipo SVD, ao tempo em segundos etc. Em geral, adotaremos como salvaguardas para interromper o método um limite máximo de iterações ( $M_{it}$ ), de SVDs ( $M_{SVD}$ ) e de passos na busca ( $M_{itLS}$ ) realizada, os quais tipicamente não são atingidos.

## Determinação dos Valores dos Parâmetros

Cada um dos três grupos de parâmetros tem sua importância no esquema algorítmico. No entanto, os parâmetros que regem a continuação homotópica são os mais desafiadores porque sobre eles há consideravelmente menos estudos com sugestões de valores e estratégias do que sobre os demais. Além disso, são muitas as combinações de sequências plausíveis de serem adotadas. Sendo assim, fixaremos os parâmetros de busca linear com base em valores encontrados na literatura para métodos semelhantes e estabeleceremos os parâmetros dos testes de parada de forma condizente com a precisão desejada. Já para os parâmetros de homotopia, nossa

abordagem inicial será a de testá-los com as instâncias aleatórias em alguns cenários distintos de configuração, de forma a obter subsídios para conjecturas sobre seus impactos no desempenho do SPGM e, conseqüentemente, a fazer escolhas mais bem fundamentadas. Para organizar melhor as ideias, separamos as configurações testadas em duas baterias de testes.

### Bateria 1: Ausência de Homotopia

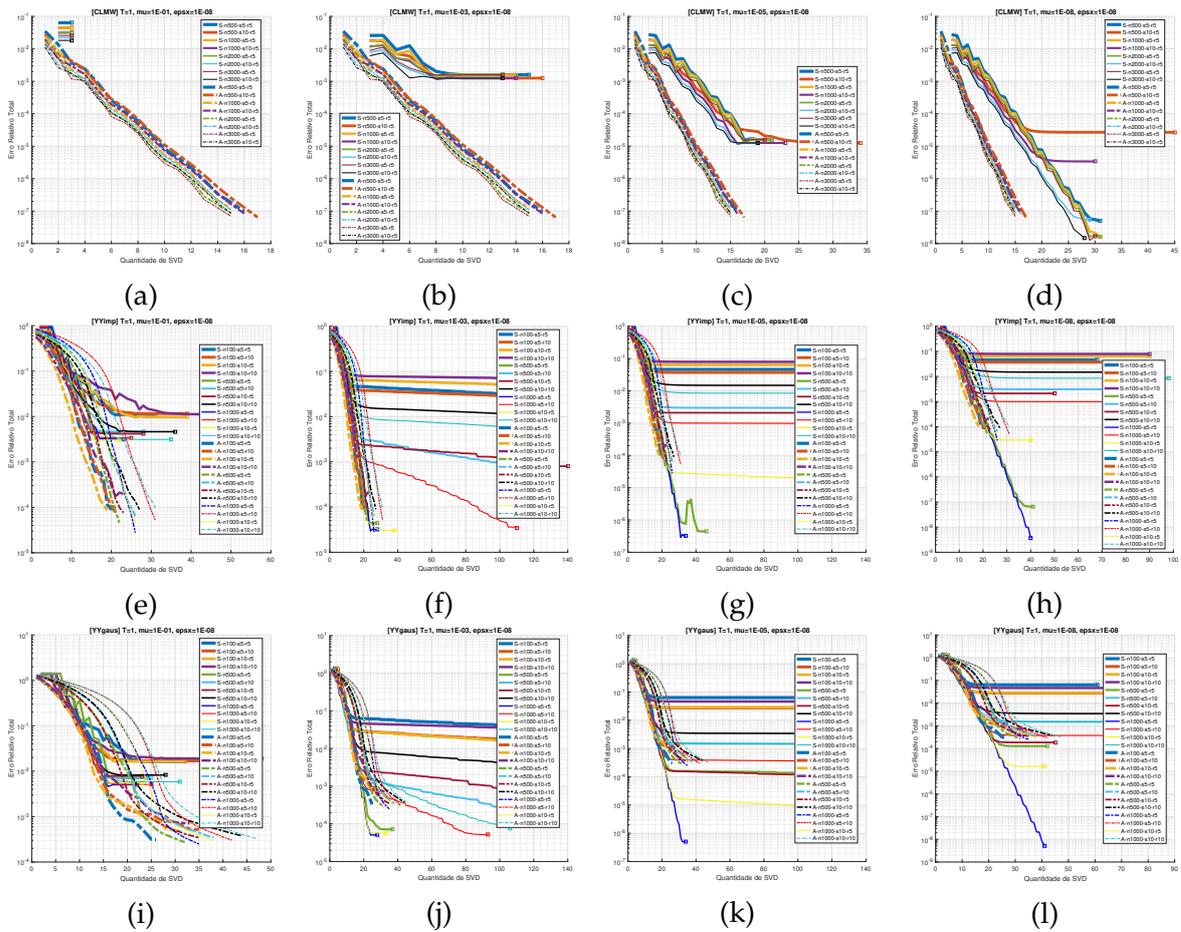
Na primeira bateria de testes de configuração, executamos o SPGM nos problemas aleatórios com  $T = 1$ , de forma que o parâmetro de suavização  $\mu$  é único ao longo das iterações de cada execução. Note que isso equivale a inibir a homotopia. Testamos oito valores diferentes para  $\mu$ , com a configuração exibida na Tabela 7, e apresentamos na Figura 6 os erros relativos em termos dos gastos em SVD.

Tabela 7 – Parâmetros do SPGM: Bateria 1 (Ausência de Homotopia).

$T = 1$	$\mu \in 10^{\{-8, -7, \dots, -1\}}$	$M_{it} = 100$	$\sigma = 10^{-4}$	$L^{(0)} = 0_{n \times n}$
$\epsilon_x = 10^{-8}$	$\alpha_{min} = 10^{-30} = 1/\alpha_{max}$	$M_{itLS} = 20$	$\zeta = 0.5$	$M = 20$

A Figura 6 ilustra os resultados de erro relativo em termos do gasto em SVDs para cada grupo de instâncias aleatórias, tanto com o SPGM quanto com o ADMM, para fins de comparação. Dos oito valores de  $\mu$  testados, selecionamos para esta exibição os seguintes:  $10^{-1}$ ,  $10^{-3}$ ,  $10^{-5}$  e  $10^{-8}$ . Nitidamente, para ambos métodos, o comportamento das curvas com as instâncias de CLMW é distinto daqueles com as de YYimp e YYgaus. Vejamos alguns destaques para o que ocorre com o SPGM.

Figura 6 – Gráficos semi-logarítmicos de erro relativo por quantidade de SVDs resultantes da primeira bateria de testes de configuração. No caso do SPGM, a homotopia foi inibida ( $T = 1$ ) e a parada foi ajustada para ocorrer quando o tamanho relativo do passo no domínio ficasse abaixo da tolerância  $\epsilon_x = 10^{-8}$ . As curvas do ADMM, foram obtidas a partir das configurações de validação do método (2). Cada linha se refere a um grupo de instâncias aleatórias: CLMW, YYimp e YYgaus, de cima para baixo. Cada coluna retrata a escolha de um valor para  $\mu$ :  $10^{-1}$ ,  $10^{-3}$ ,  $10^{-5}$  e  $10^{-8}$ , da esquerda para a direita.



Analisando cada linha, podemos observar o efeito da diminuição do valor de  $\mu$  sobre os erros relativos de cada grupo de instâncias. Para as instâncias de CLMW, da Subfig. 6(a) à Subfig. 6(d), o SPGM foi capaz de conduzir os erros relativos totais a valores da ordem de  $\mu$ . A exceção, além dos *outliers* de características  $(n, spr, rkr) \in \{(500, 10, 5), (100, 10, 5)\}$  quando  $\mu = 10^{-8}$  na Subfig 6(d), foi o caso em que  $\mu = 10^{-1}$ , para o qual os pontos iniciais já conduziram a erros abaixo do valor desse parâmetro.

Essa associação direta entre  $\mu$  e o erro relativo não se mostrou amplamente válida para as instâncias de YYimp e YYgaus, embora ainda assim possamos extrair algumas informações importantes nesse sentido. Em ambos os grupos, a dificuldade em conquistar mais precisão apesar da diminuição de  $\mu$  sugere uma limitação do

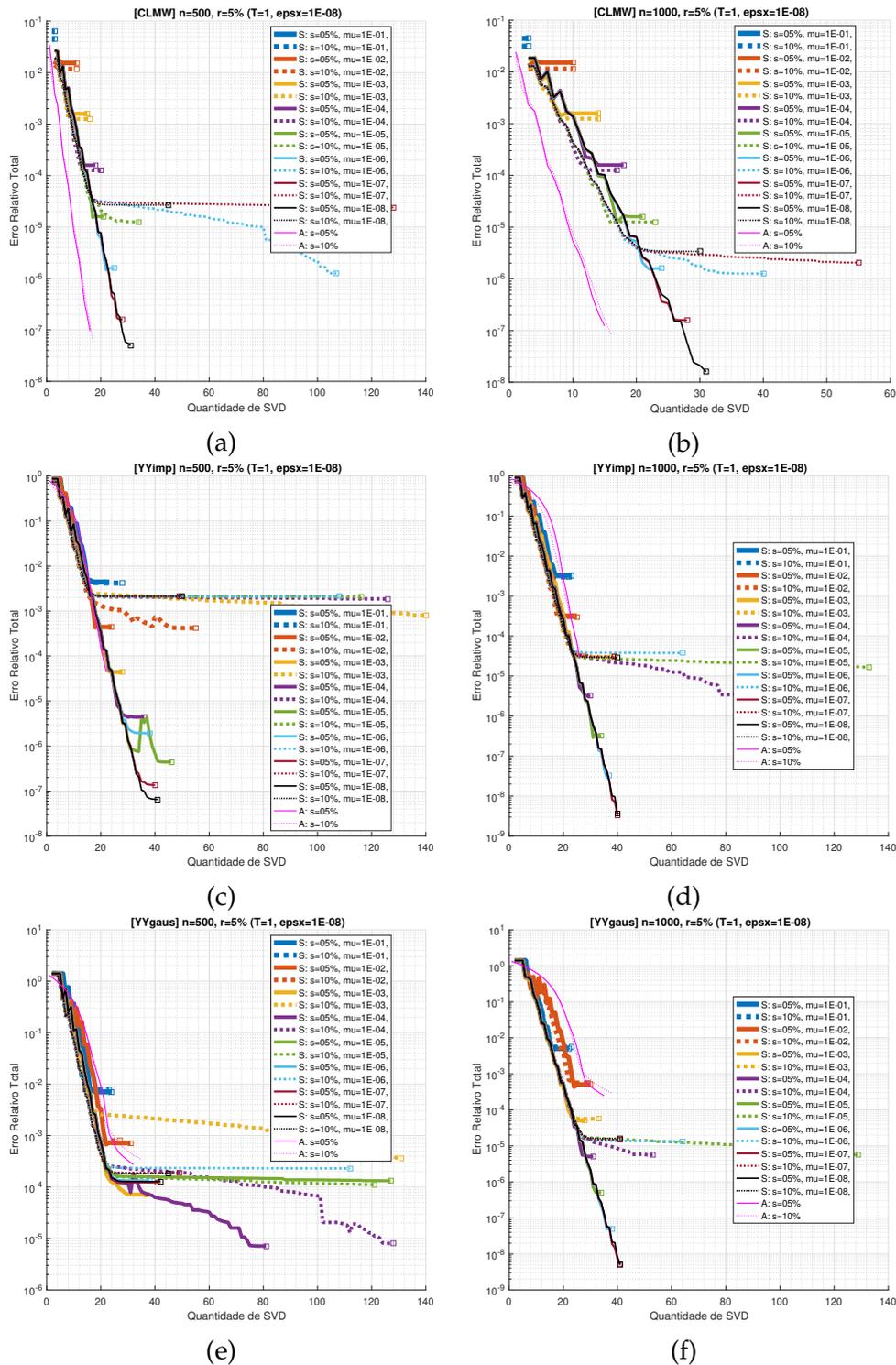
método – ou, simplesmente, das configurações adotadas – quando o SPGM é aplicado a esses tipos de estruturas matriciais. No entanto, é possível observar uma tendência que já aparecia com os problemas de CLMW: fixados a esparsidade e o posto, o SPGM tem melhor desempenho conforme a dimensão do problema aumenta. Além disso, podemos ver que sistematicamente as instâncias com  $n = 100$  são as que resultam em maior erro em todos os três grupos; já com  $n = 1000$ , com poucas exceções, o SPGM obteve os melhores de seus resultados.

Uma possível explicação para o fato de que o SPGM obteve melhores resultados nos problemas com dimensão maior é intrínseca à característica espectral do método. Lembre-se que podemos interpretar a estratégia espectral de forma que uma das etapas seja a aproximação de uma matriz hessiana de um modelo quadrático por uma matriz escalar cujo valor é, por si próprio, um representante dos autovalores da hessiana. Sob esse aspecto, quanto maior a dimensão do problema, mais autovalores a serem aproximados por esse escalar. Dessa forma, a aproximação estaria menos suscetível a *outliers*, supondo que haja na distribuição desses valores uma tendência de convergência, o que poderia garantir um pouco mais de estabilidade e, conseqüentemente, melhor desempenho.

Com relação às curvas de erro obtidas pelo ADMM – as quais são repetidas nos gráficos de uma mesma fileira horizontal, já que não há mudanças na configuração desse método –, constatamos um comportamento mais uniforme na resolução das instâncias de cada grupo. Em que pese essa consistência, vale observar que, em contraste com o SPGM, o ADMM mostrou regularmente mais dificuldade nos problemas de maior dimensão nos grupos YYimp e YYgaus. É importante ter bom desempenho com instâncias maiores, já que tipicamente são esses tipos de matrizes os que surgem na prática, como veremos nos experimentos com vídeos de vigilância.

Ainda da primeira bateria de experimentos, selecionamos agora quatro instâncias de cada grupo para que possamos olhar sob outro ângulo os efeitos da variação de  $\mu$ . A seleção corresponde às instâncias dos três grupos com características em comum:  $\{(n, spr, rkr) \mid n \in \{500, 1000\}, spr \in \{5\%, 10\%\}, rkr = 5\%\}$ . A Figura 7 é formada pelos gráficos das curvas de erro, como sempre em termos do número de SVDs, referentes a essas instâncias e associados por dimensão e grupo a que pertencem.

Figura 7 – Gráficos semi-logarítmicos de erros relativos por quantidades de SVD para as instâncias com posto percentual  $r = 5\%$ , com dimensões  $n = 500$  (à esquerda) e  $n = 1000$  (à direita), resultantes da primeira bateria de testes de configuração, realizados com o SPGM sem homotopia ( $T = 1$ ) para cada um dos diferentes valores de  $\mu$  no conjunto  $10^{-1}, 10^{-3}, 10^{-5}$  e  $10^{-8}$ . São exibidas também as respectivas curvas obtidas com os resultados do ADMM sob a configuração de validação (Tabela 2). Cada linha se refere a um grupo de instâncias aleatórias: CLMW, YYimp e YYgaus, de cima para baixo.



Com essa figura, conseguimos ver que, em geral, o ritmo de decrescimento do erro relativo conquistado pelo SPGM em cada instância não varia muito com a diminuição do valor de  $\mu$ , tendendo em todos os casos retratados a formar uma reta até o momento em que esgota a precisão alcançável atrelada à precisão da suavização em si – isto é, a  $\mu$ .

Os gráficos corroboram que, para várias instâncias, é possível estabelecer uma relação direta entre o valor de  $\mu$  e o erro relativo atingível, não apenas para CLMW mas também para YYimp e, em menor grau, até mesmo para YYgaus. Essa afirmação é ainda mais acertada quando olhamos para os gráficos da coluna à direita, com  $n = 1000$ , instâncias de tamanho maior com as quais o SPGM aparentemente lida melhor.

Em comparação com os erros obtidos com o ADMM, novamente constatamos uma contraposição com o desempenho do SPGM. O SPGM obteve erros menores do que os do ADMM, gastando inclusive menos SVDs, nas instâncias gaussianas das Subfiguras 7(e) e 7(f). O contrário ocorreu nas instâncias impulsivas de *Candès et al.* (Subfiguras 7(a) e 7(b)), sendo que até onde o ADMM é executado podemos considerar um empate nas instâncias impulsivas de Yuan e Yang (Subfiguras 7(c) e 7(d)).

Fica evidente, como esperado, que a análise de desempenho dos métodos precisa levar em consideração a precisão almejada. Veremos mais adiante que nas aplicações envolvendo imagens, um bom palpite para a precisão alvo é o de erro relativo de  $10^{-4}$ . Embora o SPGM sem homotopia tenha mostrado que pode atingir esse patamar em vários casos, a configuração do processo de continuação pode trazer mais uniformidade nos resultados, em particular nos casos das instâncias de Yuan e Yang. Como a configuração dos parâmetros de homotopia não é trivial, exibiremos a seguir os resultados de três configurações que de certa forma representam algumas possibilidades de escolha – novamente, sem pretender esgotá-las.

## Bateria 2: Três estratégias-teste

Já na outra bateria de testes, também com as instâncias aleatórias, executamos o SPGM com três estratégias diferentes para a homotopia, sendo os demais parâmetros herdados da Tabela 7:

- |   |  |   |
|---|--|---|
| <ul style="list-style-type: none"> <li>• Estratégia 1</li> <li><math>T = 4</math></li> <li><math>\mu_1 = 10^{-1}</math></li> <li><math>\mu_T = 10^{-4}</math></li> <li><math>\epsilon_x = 10^{-6}</math></li> </ul> | <ul style="list-style-type: none"> <li>• Estratégia 2</li> <li><math>T = 6</math></li> <li><math>\mu_1 = 10^{-1}</math></li> <li><math>\mu_T = 10^{-6}</math></li> <li><math>\epsilon_x = \epsilon_x(j) = \frac{\mu_j}{10}</math></li> </ul> | <ul style="list-style-type: none"> <li>• Estratégia 3</li> <li><math>T = 4</math></li> <li><math>\mu_1 = 10^{-3}</math></li> <li><math>\mu_T = 10^{-6}</math></li> <li><math>\epsilon_x = 10^{-6}</math></li> </ul> |
|---|--|---|

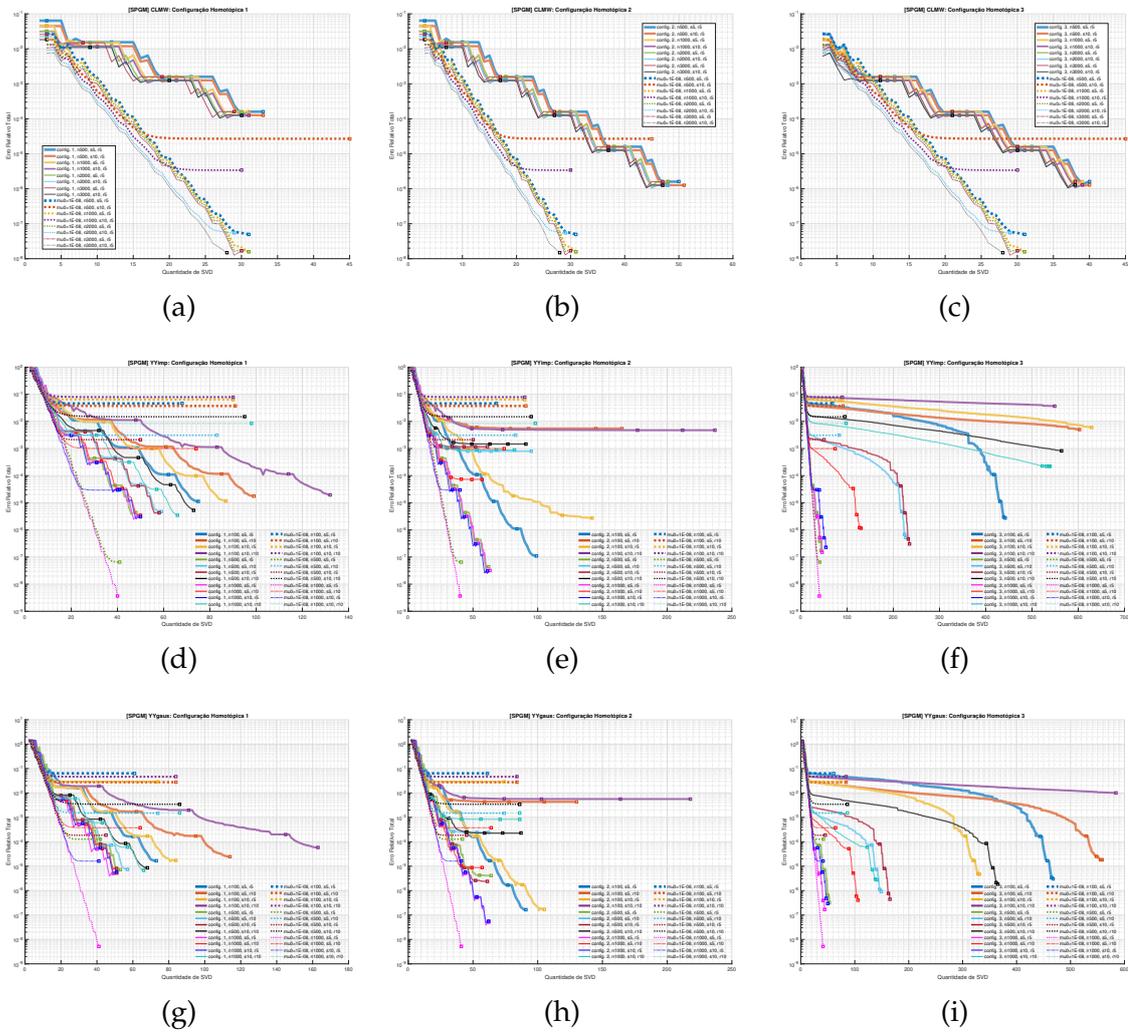
Essas estratégias foram idealizadas a partir de experimentos preliminares e não têm como objetivo esgotar as possibilidades ou trazer dados definitivos sobre a configuração do processo homotópico. Em todas elas, a sequência de parâmetros de suavização é caracterizada pela divisão por dez a cada troca de iteração homotópica; portanto, bastam  $T$  e  $\mu_1$  para obtê-las por completo. Em outro contexto, essa foi a tática adotada por Fountoulakis e Gondzio para a determinação de sua sequência de parâmetros ([39]). As estratégias 1 e 3 operam sob tolerância  $\epsilon_x$  constante e com o mesmo número de iterações homotópicas, mas diferem na sequência ( $\mu_k$ ) de forma que a primeira é mais conservadora do que a segunda na busca por precisão; ambas as estratégias 1 e 2 se iniciam com  $\mu_1 = 10^{-1}$  porém a segunda realiza duas iterações homotópicas a mais que a primeira, chegando a  $\mu_6 = 10^{-6}$ , ao mesmo tempo que adota um critério de transição de iterações homotópicas em termos do valor corrente de  $\mu_k$ . Sabemos dos testes preliminares que se a quantidade  $T$  de iterações homotópicas for grande, a tendência é que se gaste mais do que o necessário, enquanto que se  $T$  for pequeno retornaremos às questões do primeiro grupo de testes, quando  $T = 1$ .

A Figura 8 ilustra os resultados das três configurações, assim como suas comparações com o caso sem homotopia ( $T = 1$ ) e  $\mu = 10^{-8}$ , sendo cada linha referente a um grupo de instâncias: CLMW, YYimp e YYgaus, nessa ordem, de cima para baixo. As curvas sem homotopia servem também de referencial para os experimentos, já que cada uma delas é repetida nos gráficos da mesma linha. Analisemos os gráficos por grupo de instâncias (linhas).

Para as instâncias de CLMW, fica bastante evidente que em quase todos os casos os melhores desempenhos ocorrem na ausência de homotopia. Por outro lado, a presença dessa estratégia foi capaz de neutralizar o comportamento destoante das curvas associadas às instâncias ( $n$ ,  $spr$ ,  $rkr$ ) iguais a (500, 10, 5) e (1000, 10, 5) nas três configurações, o que é um indicativo de que a homotopia também aporta em robustez.

A potencialidade da homotopia para o SPGM se torna ainda mais patente quando olhamos para os grupos YYimp e YYgaus. Apesar do gasto sobressalente que a homotopia acarretou com o grupo CLMW, nesses dois outros grupos o fenômeno não foi observado: mesmo quando terminam estagnadas, o patamar de erro das curvas obtidas com homotopia é em geral melhor do que aquele gerado sob sua ausência (veja as Subfiguras 8(e) e 8(h)). Já na Configuração 1 com YYimp e YYgaus (Subfiguras 8(d) e 8(g)), as curvas de erro dos testes com  $T = 1$  que estavam estagnadas com erro relativo da ordem de  $10^{-2}$  foram decrescidas consideravelmente para a ordem de  $10^{-6}$  ou  $10^{-5}$ , muitas vezes com gasto comparável à ausência de homotopia. Vale notar que, na estratégia 1, o valor final do parâmetro de suavização é  $\mu_4 = 10^{-4}$ , o que, como vimos, provavelmente está relacionado ao nível de precisão atingido. Sem homotopia, por outro lado, adotamos  $\mu = 10^{-8}$ .

Figura 8 – Gráficos semi-logarítmicos de erros relativos por quantidades de SVD de cada uma das três configurações elaboradas, em contraste com a configuração sem homotopia ( $T = 1, \mu_1 = 10^{-8}$ ). Cada coluna se refere a uma das três configurações, enquanto cada linha indica um grupo de instâncias aleatórias: CLMW, YImp e YGaus, de cima para baixo.



Com relação à escolha da tolerância  $\epsilon_x$  – entre mantê-la sempre fixada ou adaptá-la ao  $\mu$ , os testes não capturaram grandes impactos, com exceção possivelmente de que, na estratégia 2, sua política para  $\epsilon_x$  garantiu que ao menos a última iteração homotópica não fosse muito breve. Com algum cuidado com as escalas horizontais, podemos ver que em muitas instâncias o SPGM gastou um pouco menos SVDs para atingir a precisão final na estratégia 2 do que para atingir o mesmo patamar de erro na estratégia 1, em particular no casos de CLMW. Por outro lado, é possível encontrar casos em que o  $\epsilon_x$  da estratégia 2 se mostrou grande demais: a curva de YImp correspondente a  $(n, spr, rkr) = (500, 5, 10)$  (linha sólida azul claro na Subfigura 8(e)) se mantém estagnada no erro relativo da ordem de  $10^{-3}$ , enquanto que essa mesma instância na Subfigura 8(d) atingiu precisão de  $10^{-6}$  com menos iterações homotópicas,

passando mais tempo nos mesmos valores de  $\mu$  devido à escolha de  $\epsilon_x$ . Ao menos com essa mesma instância no grupo YYgaus, um comportamento similar também ocorreu. Portanto, vemos que adotar um valor pequeno para  $\epsilon_x$  pode causar estagnação precoce, mas diminuí-lo em demasia pode acarretar em desperdício de SVDs, equiparável a um *oversolving*.

Uma alternativa para a escolha da sequência de tolerâncias no tamanho relativo do passo pode ser aquela em que se adota  $\mu_k/10$ , assim como na estratégia 2, porém com salvaguardas limitantes, como, por exemplo,  $10^{-10} \leq \epsilon_x \leq 10^{-4}$ .

Uma curiosidade sobre as curvas de erro dos resultados das instâncias de CLMW é que fora dos patamares de estagnação do erro, a taxa média de decrescimento é muito semelhante à taxa de decrescimento sem homotopia. Graficamente, as curvas de erro com homotopia apresentam trechos que são aproximadamente paralelos à curva sem homotopia.

### Considerações sobre as estratégias

A determinação de qual estratégia é a mais apropriada para resolver os problemas aleatórios depende do objetivo a ser alcançado, o que engloba não apenas a precisão que se deseja, mas também as restrições orçamentárias.

Como esperado, as estratégias 2 e 3 são as únicas capazes de atingir erros relativos da ordem de  $10^{-6}$  nos três grupos testados, ainda que não o faça com todas as instâncias, mas principalmente com as que combinam dimensões maiores com posto e esparsidade menores. No entanto, há gastos proibitivos de orçamento e, considerando as instâncias em que há estagnação precoce, constatamos menos indicativos de robustez. Já a estratégia 1, construída para mirar precisões moderadas, consegue consertar os defeitos da ausência de homotopia sem precisar de um orçamento exagerado e conseguindo lidar bem com praticamente todas as instâncias de cada um dos três grupos. Nesse sentido, consideramos que foi a estratégia que se saiu melhor nos testes e, portanto, foi a que nos norteou nos demais experimentos com o SPGM.

## 4.7 Comparação de Desempenho nas Instâncias Aleatórias

Uma maneira de condensar e resumir os dados dos experimentos desta seção é usar os *data profiles*, adotando como medida de desempenho o erro relativo total e, como unidade de custo, a quantidade de SVDs. Com isso, os *data profiles* retratam o percentual de problemas resolvidos com precisão  $\tau$  – isto é, com erro relativo menor ou igual a  $\tau$  – em termos do esforço computacional medido em quantidade de SVDs realizadas.

A Figura 9 contém os *data profiles* cujo conjunto de *solvers* consiste no ADMM com configuração padrão e  $\epsilon_{\text{relinf}} = 10^{-14}$  com  $M_{it} = 1000$ , e em três versões do SPGM, aquela sem homotopia e as 3 estratégias de configuração mencionadas anteriormente. Cada fileira horizontal de gráficos se refere a um conjunto de problemas distinto; a saber, de cima para baixo: CLMW, YYimp, YYgaus e, por último, todos eles juntos.

Os *data profiles* da Figura 9 ratificam as observações que destacamos anteriormente. De um modo geral, as configurações testadas para o SPGM se mostraram mais adaptadas a um ou dois tipos de instância aleatória do que a outro(s). O SPGM sem homotopia conseguiu atingir precisão da ordem de até  $10^{-4}$  com poucas SVDs no grupo CLMW; no entanto, não apresentou bom desempenho nos outros dois grupos. Por outro lado, a estratégia de configuração 1 se manteve competitiva em todos os grupos, com a esperada exceção dos gráficos com  $\tau = 10^{-6}$ , cuja precisão não é o alvo dessa estratégia.

Um aspecto que será relevante para entendermos o desempenho dos métodos nas instâncias práticas é o de que SPGM (config1) nas instâncias de YYgaus com precisão  $\tau = 10^{-4}$  é mais eficiente do que o ADMM, isto é, atinge essa precisão em mais problemas, gastando menos SVDs. Nesse caso, ainda consegue resolver todos os problemas satisfatoriamente. Nossos questionamentos neste momento se dividem em duas direções: qual deve ser a precisão almejada e qual é o tipo de instância aleatória que melhor representa as instâncias práticas.

Uma constatação relevante a respeito do grupo CLMW é a de que as curvas dos *data profiles* referentes a ele se elevam bruscamente não só com o ADMM mas até mesmo para as versões do SPGM. Esse comportamento sugere que não há, de fato, uma diversidade entre as instâncias desse grupo no quesito dificuldade de resolução. Parece que, ao resolver uma das instâncias, as outras também são resolvidas, o que explicaria o comportamento verticalizado nas Subfiguras 9(a), 9(b) e 9(c).

Com relação à precisão de  $10^{-3}$ , podemos constatar pela Subfigura 9(j) que o SPGM-config1 ( $T = 4$ ,  $\mu \in 10^{-\{1,2,3,4\}}$  e  $\epsilon_x = 10^{-6}$ ) se mostrou robusto ao resolver todos os problemas com a precisão indicada, gastando menos do que 100 SVDs por instância. Na perspectiva da precisão de  $10^{-4}$ , ilustrada na Subfigura 9(k), apesar da indicação de que 25% dos problemas não foram resolvidos a contento, todos oriundos do grupo CLMW, podemos atestar a robustez ao perceber pela Subfigura 8(a) que em todos os problemas o método alcançou erro relativo menor do que  $1.1 \times 10^{-4}$ . Ocorre que a escolha de  $\mu_T = 10^{-4}$  nesse grupo de problemas parece funcionar como limitante inferior para o erro relativo, ainda que, para todos os efeitos práticos, não seja absurdo dizer que a precisão de  $10^{-4}$  tenha sido atingida. Portanto, avaliamos que a estratégia de configuração 1 seja robusta.

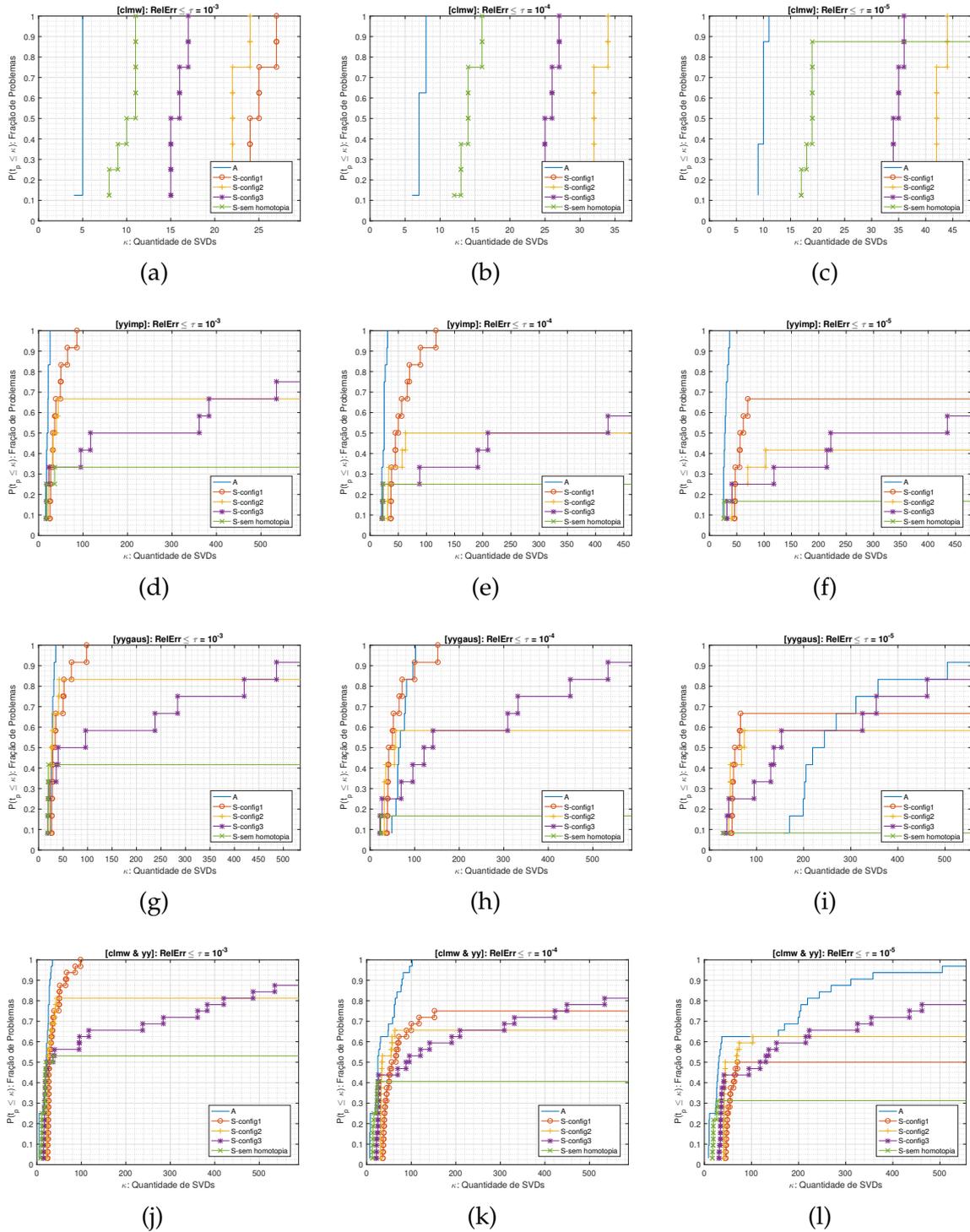
Com respeito à eficiência do SPGM-config1, é relevante fazer dois recortes,

além daquele já feito sobre a precisão-alvo: o do tipo de instância e o das dimensões. Fica evidente da Figura 8 que o grupo de problemas CLMW, de caráter impulsivo, trouxe mais dificuldade para o SPGM do que os demais grupos, ao mesmo tempo que esse grupo não comprometeu o desempenho do ADMM, o qual consistiu em excelentes resultados com poucas iterações. Entretanto, como vimos, o CLMW parece não representar de fato uma grande variabilidade de problemas. Essa suspeita surgiu tanto do desempenho de ambos os métodos nessas instâncias quanto da análise de suas estruturas de esparsidade e posto – este último que sequer sofre variação, se consideramos sua caracterização percentual, fixada em 5%.

O recorte da dimensão diz respeito à observação de que o SPGM teve mais facilidade com as instâncias aleatórias de maior dimensão do que as de pequeno porte, o que de certa forma foi oposto ao que vimos do ADMM. Se descartássemos dos grupos YYimp e YYgaus as instâncias de dimensão  $n = 100$ , o desempenho geral do SPGM teria um salto considerável. Se nos restringíssemos a  $n = 1000$ , os resultados seriam ainda mais competitivos com os do ADMM. Isso pode ser visto, por exemplo, a partir dos resultados ilustrados pela Figura 8.

Essas três questões, do tipo estrutural e da dimensão das instâncias aliadas à precisão almejada, nos levam ao ponto central da investigação dos métodos com instâncias práticas de videovigilância: há uma estrutura comum a elas, cujas dimensões tendem a ser grandes, e uma precisão-alvo que, em certo sentido, favoreçam as particularidades do SPGM em detrimento da generalidade do ADMM? Veremos no Capítulo 6, respaldados por experimentos numéricos, que a resposta a essa pergunta tende a ser afirmativa e que, ademais, as instâncias de teste se aproximam mais de YYgaus do que de CLMW.

Figura 9 – Gráficos do tipo *Data Profiles* com erro relativo como medida de desempenho sob tolerância  $\tau$  e número de SVDs como medida de custo. Os métodos comparados são quatro: o ADMM com configuração padrão,  $M_{it} = 1000$  e tolerância de infactibilidade  $\epsilon_{relinf} = 10^{-14}$  (“A”); o SPGM com as três estratégias de configuração (“config 1”, “config2”, “config3”), e a versão sem homotopia e com  $\mu = 10^{-8}$  (“sem homotopia”).



## 5 MODELAGEM DE DADOS DE INSTÂNCIAS PRÁTICAS, ESPAÇO DE EXIBIÇÃO E PRECISÃO-ALVO

Neste capítulo, explicamos como a situação prática se adéqua ao *framework* da decomposição matricial posto-esparsidade (SLRMD). Além disso, a partir do objetivo de representação visual intrínseco aos problemas com imagens, definimos o que chamamos de Espaço de Exibição, que é um conjunto discreto que, na prática, rege a demanda por qualidade das soluções. Definimos nesse espaço algumas medidas que nos auxiliam a quantificar a semelhança entre as imagens desse conjunto e, com isto, estipulamos uma precisão a ser almejada pelos métodos de otimização na resolução dos problemas com dados reais.

### 5.1 Dos vídeos aos Iterandos

Neste trabalho, o conceito de *vídeo* corresponde a uma sucessão finita de imagens de mesmo tamanho capturadas a partir de um ponto de vista fixo (ou seja, a câmera não se move durante a gravação). Por simplicidade, consideramos apenas vídeos em escala de cinza, nos quais a coloração pode variar entre preto, branco e um número predeterminado de tons intermediários entre ambos, como veremos com mais detalhes ainda neste capítulo. Além disso, embora seja comum que as imagens sigam um arranjo cronológico, a princípio não há indícios de que a ordenação dos quadros à partida seja essencial na nossa abordagem, desde que seja mantida fixa ao longo de todo o processo.

Em termos gerais, o problema de aplicação prática que abordamos trata da separação de um vídeo digital em dois outros de mesma duração, de forma que um deles retrate apenas os objetos que se mantêm em repouso ao longo da cena, enquanto o outro, complementarmente, contenha os objetos em movimento. Chamaremos de *plano de fundo* o conjunto de objetos em repouso e, em contrapartida, a coleção de todos os demais elementos da cena será simplificada e nomeada de *primeiro plano*, embora essas termos tenham significados que variam com a área do conhecimento. É importante ter em mente que a distinção entre as duas componentes é fundamentada num critério de movimentação espacial e não diz respeito, por exemplo, à distância dos objetos à câmera filmadora. O próprio termo “plano de fundo” pode se referir tanto ao vídeo quanto a alguma de suas imagens em particular, sendo a distinção

explicitada quando não for possível fazê-la a partir do contexto; o mesmo vale para “primeiro plano”. A depender do contexto, o enfoque da decomposição pode estar na reconstituição de um ou do outro. As definições precisas desses dois termos não são essenciais no escopo deste trabalho, inclusive porque há diversas situações empíricas em que a discriminação entre um e outro não é muito clara, como veremos no Capítulo 6.

Dado que queremos decompor o vídeo em uma parte contendo os elementos estáticos da cena e outra composta pelo que se movimenta, fica subentendida a existência de um cenário fixo pelo qual alguns elementos de tamanho comparativamente menor se deslocam com o tempo. Indo um pouco além, podemos assumir que esse cenário, mesmo considerado fixo, sofre pequenas alterações devidas, por exemplo, a mudança de luminosidade ou a oscilações provocadas por ventos ou intrínsecas ao funcionamento de certos objetos. Apesar da imprecisão nessa tentativa de delimitar quais aplicações são de nosso interesse, essa noção é suficiente para explorar um pouco as extensões e os limites do escopo dos problemas práticos tratados aqui. Mais do que isso, nosso interesse também está em abordar instâncias que fogem um pouco do ideal, já que de partida as hipóteses teóricas de recuperabilidade dificilmente são satisfeitas em um contexto prático como o dos vídeos.

### Posto Baixo e Esparsidade

Para adquirir intuição sobre a transição entre o problema de aplicação e o modelo matemático, assumamos por ora que o procedimento de decomposição do vídeo tenha sido realizado com sucesso e que tenhamos em mãos um vídeo do plano de fundo e outro do primeiro plano. Então, as imagens que compõem o plano de fundo tendem a ter grande semelhança entre si, embora dificilmente sejam idênticas devido às possíveis variações no ambiente que admitimos anteriormente. Portanto, faz sentido pensar que é possível eleger e fixar um conjunto com algumas poucas imagens de forma que seja plausível representar o plano de fundo em qualquer instante a partir de combinações *básicas* das imagens selecionadas. E é fazendo a cardinalidade desse conjunto de imagens corresponder ao posto de uma matriz que associamos o plano de fundo à componente de posto baixo do problema SLRMD.

Por outro lado, as informações essenciais de cada imagem que integra o vídeo do primeiro plano, com os objetos em movimento, tipicamente ocupam apenas uma pequena fração da área total da imagem, sendo aquele espaço originalmente ocupado pelo plano de fundo preenchido por uma cor que representa a ausência de informação visual, de luminosidade: o preto. Por intermédio dessa característica de ausência de luminosidade é que associaremos o vídeo do primeiro plano à componente esparsa do modelo matemático.

## Redimensionamento e Conversão

Com o intuito de investigar mais profundamente a aplicação prática e a adequação ao modelo matemático, é necessário compreender os fundamentos computacionais do processamento de imagens. Para o leitor menos familiarizado com o tema, o Apêndice E aborda os conceitos essenciais para a compreensão do que se discute a partir deste ponto.

Como sabemos, imagens podem ser armazenadas computacionalmente como matrizes numéricas. Já um vídeo, enquanto sequência de imagens, pode ser manipulado para formar uma matriz bidimensional se reorganizamos os elementos de cada uma de suas imagens como um vetor coluna, emparelhando tais vetores sucessivamente um ao lado do outro, como ilustra a Figura 10. Respeitando-se a ordenação dos elementos, esse processo é nitidamente reversível.

Assumimos que estamos sob as condições gerais da aplicação de videovigilância e queremos decompor um vídeo digital  $\mathcal{V}$  de 8 bits em outros dois de mesmas características,  $\mathcal{V}_1$  e  $\mathcal{V}_2$ , de forma que o primeiro contenha as informações de primeiro plano e o segundo, as do plano de fundo.

Supondo que  $\mathcal{V}_2$  apresenta poucas variações entre seus *frames* e que, numericamente, há certa dependência linear entre eles, podemos construir uma matriz  $L$  em que cada coluna é um *frame* do vídeo reorganizado usando a indexação linear de seus elementos, na ordem padrão do MATLAB (cf. (E.1)). Dessa forma, é de se esperar que  $L$  seja uma matriz de posto baixo.

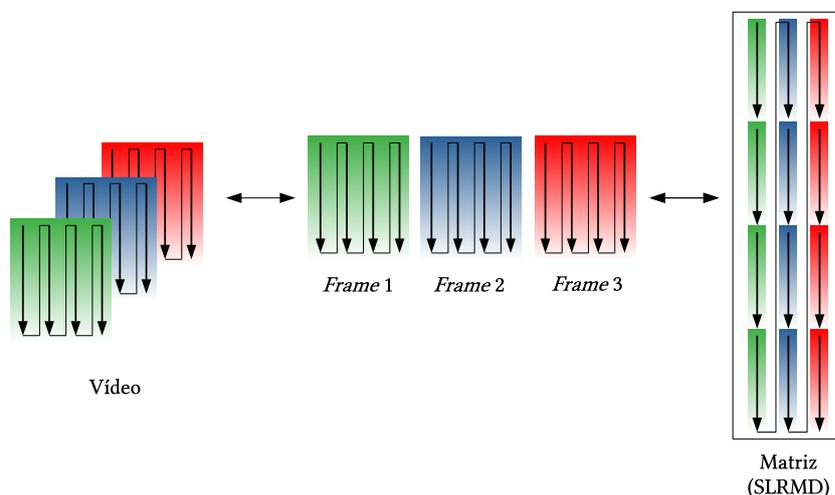
Paralelamente, admitindo que os objetos móveis em  $\mathcal{V}_1$  ocupam a menor parte dos píxeis do vídeo, podemos construir uma matriz  $S$  cujas colunas são os *frames* de  $\mathcal{V}_1$  reorganizados de maneira compatível com o que foi feito com  $\mathcal{V}_2$ . Sendo assim, a matriz  $S$  terá em grande parte de suas entradas o valor zero e será considerada esparsa.

Decorre dessas implicações que, se o vídeo original for reestruturado de forma consistente como uma matriz  $A$ , é razoável tentar separá-la em suas componentes complementares  $S$  e  $L$  por meio do problema de otimização (2.5). Obviamente, o que não é razoável de se esperar é que a matriz  $A$  satisfaça as condições teóricas de separação e recuperabilidade. Na prática, no entanto, bons resultados têm sido reportados com o uso dessa estratégia, como mostraram Candès *et. al.* [21].

### Tipos de Dados na Conversão

Faz parte da adequação ao modelo matemático adaptar os diferentes tipos de dados envolvidos no processo. Os vídeos serão carregados a partir de *frames* armazenados como imagens em arquivos separados. Nas situações em que os vídeos

Figura 10 – Ilustração da correspondência entre os elementos dos *frames* do vídeo e os da matriz do problema de decomposição. As setas indicam a ordem percorrida na reorganização dos elementos.



sejam originalmente disponibilizados em cores, realizaremos a conversão dos *frames* para o formato monocromático em escala de cinza, ainda em `uint8`. Em todos os casos, no MATLAB, converteremos as matrizes com os dados de entrada do tipo `uint8` para `double`, reescalando os valores para o intervalo  $[0, 1]$ , por precisão e conveniência.

Nesta primeira adaptação, os erros que podem surgir são meramente de arredondamento, e a transformação é injetora. Iniciado o processo de otimização, as matrizes envolvidas passam a poder admitir como entradas quaisquer valores reais (ponto flutuante de precisão dupla), grandes ou pequenos, positivos ou negativos. Ao fim do processo, quer o intuito seja exibir as imagens, quer seja armazená-las, será necessário converter os valores de volta para o formato de 8 *bits*. Diferentemente da primeira conversão, essa última não é injetora, já que há apenas 256 valores possíveis para cada entrada.

Essas observações são relevantes não apenas porque envolvem uma decisão sobre o que fazer com os valores advindos da otimização que extrapolem o intervalo unitário padrão, mas também porque nos ajudam a entender que é possível não chegar no valor exato, a nível de otimização, mas já ter atingido a imagem final almejada, a nível de exibição e armazenamento. A distinção entre o que ocorre a nível de otimização e o reflexo de seus resultados na exibição das imagens torna propícia a definição de espaços associados a cada um desses níveis, como discutimos a seguir.

## 5.2 Espaços de Otimização e de Exibição

Qualquer cálculo numérico realizado no computador por meio da aritmética de ponto flutuante está sujeito a erros de aproximação decorrentes de uma limitação tecnológica em representar os números reais; para aproximar os elementos desse conjunto infinito, dispomos apenas de um conjunto finito de números em ponto flutuante, que nada mais é do que um subconjunto dos números racionais. Naturalmente, essa capacidade de representação tende a aumentar com o avanço da tecnologia.

Ao lidar com imagens, acrescentam-se ainda outras complicações tecnológicas de captação (aquisição) e processamento. No caso de sistemas de segurança, a adoção de câmeras de videovigilância que captam apenas imagens monocromáticas se justifica, dentre outros fatores, devido ao custo e à disponibilidade de bons equipamentos e à capacidade de processamento e armazenamento dos conteúdos gravados ao longo do tempo.

Por outro lado, há situações em que a demanda por qualidade nas imagens é baseada na capacidade visual humana de distinguir os fenômenos que são retratados em vídeo. Dependendo do contexto, a quantidade de tons de cinza que o olho humano típico consegue distinguir pode tornar desnecessária uma escala de cor muito detalhista. Embora o olho humano típico consiga perceber uma enorme faixa de intensidades de luz, ele não é capaz de lidar com todas elas ao mesmo tempo, já que adapta sua sensibilidade para captar apenas um trecho consideravelmente menor da faixa total [42, Seção 2.1, *Brightness Adaptation and Discrimination*].

Em geral, não é possível estabelecer valores exatos para a quantidade mínima de tons de cinza suficiente para que o olho humano já não consiga fazer plena distinção entre as cores. O motivo é que a visão humana se apoia não apenas em um determinado ponto enfocado da imagem mas também em características dos arredores e em sua variação com o tempo, se houver, e ainda está sujeita aos ruídos inerentes às imagens, o que torna essa análise muito mais complexa.

Ainda assim, podemos mencionar duas experiências ilustrativas sobre a quantidade mínima de tons de cinza que contempla, sem folgas, a nossa percepção visual, ambas extraídas de [42]. Na primeira, uma pessoa é colocada para contar a quantidade de mudanças na intensidade percebida (brilho), sob certas condições, quando uma fonte de luz varia sua intensidade do mínimo ao máximo. Segundo os autores, um observador típico é capaz de distinguir aproximadamente uma ou duas dezenas de variações e, *grosso modo*, isso tem a ver com a quantidade de intensidades que uma pessoa consegue observar em um ponto qualquer de uma imagem monocromática [42, p. 52]. Entretanto, como alertam os próprios autores, isso não significa que uma imagem possa ser bem representada por uma escala tão escassa assim. O que se

afirma é que se a quantidade de intensidades for menor do que algo em torno de 20, certamente deve se esperar que ao menos problemas de contorno sejam observados na imagem monocromática.

A segunda experiência está no Exemplo 2.3 da mesma obra, que aborda os efeitos da variação do número de níveis de intensidade em uma imagem digital. Gonzalez e Woods exibem e comparam oito versões de uma mesma imagem de tomografia computadorizada, em cada uma mantendo fixada a resolução em píxeis e fazendo variar a quantidade total de níveis de cinza na escala de cor entre os valores 256, 128, 64 e 32. Sua conclusão é de que, *grosso modo*, imagens de  $256 \times 256$  píxeis, impressas em tamanho  $5 \times 5$  cm e com 64 tons de cinza são as imagens com menor resolução espacial e de intensidade que podemos esperar que sejam razoavelmente livres de problemas de falso contorno e distorções visíveis [42, p. 74]

Esses e outros fatores culminam numa importante característica presente na situação prática que este trabalho se propõe a abordar, a qual envolve o uso de imagens monocromáticas de 8 *bits* que são convertidas e associadas a iterandos de métodos de otimização contínua. No fim das contas, as imagens resultantes do processo de otimização serão exibidas ou armazenadas sob as condições que mencionamos, de forma que é conveniente que haja uma sintonia entre a precisão que se exige do método e a precisão advinda do final da representação da solução. Por conta do contexto prático, almejar altas precisões com o método de otimização pode ser dispensável.

Concretamente, o conjunto de imagens digitais de 8 *bits* com resolução de  $w \times h$  píxeis tem colossalmente menos elementos do que o conjunto de matrizes de mesma dimensão em ponto flutuante de precisão dupla (64 *bits*). Afinal, apenas a nível de píxel, com valores escalados para o intervalo  $[0, 1]$ , temos<sup>1</sup> aproximadamente  $2^{52} \cdot \frac{1}{2} 2^{11} \approx 4.6 \times 10^{18}$  possibilidades de valores não uniformemente espaçados contra  $2^8 = 256$  números igualmente distanciados entre si.

Com tudo isso, entendemos que é fundamental distinguir o conjunto das matrizes cujas entradas assumem exatamente o valor das intensidades de cor da escala de cinza de 8 *bits*, definindo o que chamaremos de Espaço de Exibição, em contraposição ao espaço vetorial  $\mathbb{R}^{m \times n}$ , de otimização. A ideia é estabelecer medidas que captem as características próprias de cada um desses espaços e que nos auxiliem a comparar o desempenho dos métodos de otimização no contexto das aplicações.

<sup>1</sup> Aproximação obtida desprezando os números subnormais e assumindo o padrão que divide os 64 *bits* entre o sinal (1), a mantissa (52) e o expoente (11), sendo apenas metade dos expoentes reservada para os números no intervalo  $[0, 1]$ ; um valor mais preciso deveria considerar que dos  $2^{11}$  expoentes, dois são reservados para casos especiais, a metade é para expoentes positivos e um deles é o expoente nulo, totalizando  $2^{52} \cdot \left[ \frac{1}{2} (2^{11} - 2) - 1 \right]$ ; a inclusão dos extremos do intervalo, 0 e 1, não altera a aproximação.

**Definição 11 (Espaços de Exibição).** Chamaremos de *Espaço de Exibição* ( $\mathbb{E}$ ) o conjunto de todas as matrizes numéricas que representam exatamente imagens digitais de determinada dimensão em píxeis. Em particular, como estamos interessados nas imagens digitais monocromáticas de 8 *bits*, se fixamos uma resolução de  $m \times n$  píxeis então o conjunto  $\mathbb{E}$  é composto pelas matrizes  $m \times n$  com entradas inteiras entre 0 e 255, com extremos incluídos, isto é:

$$\mathbb{E} = \{0, 1, \dots, 255\}^{m \times n}. \quad (5.1)$$

Quando os valores precisarem estar restritos ao intervalo  $[0, 1]$ , podemos reescalar os valores e obter o que chamaremos de *Espaço de Exibição Normalizado* ( $\mathbb{E}_{[0,1]}$ ):

$$\mathbb{E}_{[0,1]} = \left\{0, \frac{1}{255}, \frac{2}{255}, \dots, \frac{254}{255}, 1\right\}^{m \times n}. \quad (5.2)$$

Por abuso de terminologia, podemos fazer referência aos Espaços de Exibição mesmo quando as dimensões das matrizes em questão diferirem entre si, sem prejuízo da compreensão.

### Conversão para Exibição

A conversão de uma matriz  $X \in \mathbb{R}^{m \times n}$  do Espaço de Otimização para o Espaço de Exibição  $\mathbb{E}$  pode ser feita em duas etapas: primeiro, descartamos os sinais das entradas matriciais (ou seja, consideramos seu valor absoluto) e reescalamos os valores para o intervalo  $[0, 1]$ ; por último, modificamos suas entradas para que assumam valores apenas entre os primeiros 256 inteiros não negativos. Nesse processo, ocorre também a alteração do tipo de dado da matriz inicial, que passa de `double` para `uint8` (inteiros de 8 *bits* sem sinal)<sup>2</sup>.

Procedemos a seguir com a formulação matemática dessas duas etapas em que consiste a conversão. Para facilitar a compreensão da expressão final, introduzimos brevemente algumas notações preliminares.

Considere os conjuntos  $[m]$  e  $[n]$ , respectivamente definidos por

$$[m] = \{1, 2, \dots, m\} \quad \text{e} \quad [n] = \{1, 2, \dots, n\},$$

em que  $m$  e  $n$  são inteiros positivos. Sejam  $B = (b_{ij})$  uma matriz de dimensão  $m \times n$  e  $B^{|\cdot|} = (b_{ij}^{|\cdot|})$  uma matriz de mesma dimensão tal que

$$b_{ij}^{|\cdot|} = |b_{ij}| \quad \text{para todo } i \in [m], j \in [n], \quad (5.3)$$

<sup>2</sup> No MATLAB, essa tarefa pode ser realizada combinando as funções `uint8` e `mat2gray`, mas esta última só está disponível no *Image Processing Toolbox*<sup>TM</sup>, do qual não fazemos uso neste trabalho.

isto é,  $B^{|\cdot|}$  é a matriz em que cada elemento é igual ao valor absoluto do elemento correspondente em  $B$ . Sejam, ainda, os operadores  $\min$  e  $\max$  aqueles que retornam, respectivamente, os valores mínimo e máximo da matriz sobre o qual operam, isto é,

$$\min(B) \stackrel{\text{def}}{=} \min_{i \in [m], j \in [n]} \{b_{ij}\} \quad \text{e} \quad \max(B) \stackrel{\text{def}}{=} \max_{i \in [m], j \in [n]} \{b_{ij}\}. \quad (5.4)$$

Além disso, adotemos  $\mathbb{1}_{m \times n}$  como a matriz de dimensão  $m \times n$  cujos elementos são todos iguais a 1. Por fim, seja  $\lfloor B \rfloor$  a matriz das entradas de  $B$  arredondadas para o inteiro mais próximo, com desempate para longe do zero; ou seja, se  $\lfloor B \rfloor = (\beta_{ij})$ , então

$$\beta_{ij} = \begin{cases} \left\lfloor b_{ij} + \frac{1}{2} \right\rfloor, & \text{se } b_{ij} \geq 0, \\ \left\lceil b_{ij} - \frac{1}{2} \right\rceil, & \text{caso contrário,} \end{cases} \quad (5.5)$$

em que  $\lfloor x \rfloor$  representa o chão de  $x$  (o maior inteiro menor ou igual a  $x$ ) e  $\lceil x \rceil$  representa o teto de  $x$  (o menor inteiro maior ou igual a  $x$ ). Essa estratégia de arredondamento é a adotada pelo MATLAB em sua função nativa `round` com um argumento de entrada e, portanto, em valor,  $\lfloor B \rfloor$  é precisamente igual a `round(B)`.

Com essa notação, e supondo que  $X^{|\cdot|}$  não é uma matriz constante, podemos representar a primeira etapa da seguinte maneira:

$$\widehat{X} = \frac{1}{\max(X^{|\cdot|}) - \min(X^{|\cdot|})} \cdot (X^{|\cdot|} - \min(X^{|\cdot|}) \cdot \mathbb{1}_{m \times n}) \in [0, 1]^{m \times n}. \quad (5.6)$$

A raríssima situação em que  $X^{|\cdot|} = \alpha \cdot \mathbb{1}_{m \times n}$ , para algum  $\alpha > 0$ , poderia ser tratada separadamente, a depender do valor da constante, mas não nos deteremos nisso. Observe que a conversão em (5.6) força a matriz  $\widehat{X}$  a ter ao menos um 0 e um 1 entre seus elementos, e esta não é a única maneira de converter as entradas da matriz original  $X$  ao intervalo unitário.

Por sua vez, a segunda e última etapa da obtenção da imagem  $X_I$  parte de  $\widehat{X} \in [0, 1]^{m \times n}$  e pode ser descrita matematicamente como segue:

$$X_I = \left\lfloor 255 \cdot \widehat{X} \right\rfloor. \quad (5.7)$$

Computacionalmente, no MATLAB, o resultado da Equação (5.7) pode ser calculado pelo comando `round(255 *  $\widehat{X}$ )`.

Em certos momentos, será útil destacar as matrizes resultantes da conversão ao Espaço de Exibição. Com essa finalidade, introduzimos uma notação especial para o operador de conversão.

**Definição 12 (Operador de Exibição).** Chamaremos de Operador de Exibição  $\mathcal{I}$  aquele que converte uma matriz numérica  $X \in \mathbb{R}^{m \times n}$  ao Espaço de Exibição, resultando em

uma imagem digital  $X_I$ :

$$\begin{aligned} \mathcal{I} : \mathbb{R}^{m \times n} &\rightarrow \mathbb{E}, \\ \mathcal{I}(X) &= X_I, \end{aligned} \tag{5.8}$$

em que  $X_I$  é a matriz calculada pelo processo sintetizado pelas Equações (5.6) e (5.7).

**Observação.** Note que uma consequência da definição do efeito do operador  $\mathcal{I}$  é que o intervalo  $[0, 1]$  é particionado em 255 subintervalos, sendo que os subintervalos que após a conversão conduzem aos inteiros 0 e 255 têm metade do comprimento dos demais. O subintervalo que conduz ao 0 é o  $\left[0, \frac{1}{2} \cdot \frac{1}{255}\right]$ , enquanto o subintervalo que conduz ao 255 é o  $\left[\left(\frac{1}{2} + 254\right) \frac{1}{255}, 1\right]$ , cada um com comprimento  $\frac{1}{2} \cdot \frac{1}{255} = \frac{1}{510}$ . Todos os outros subintervalos têm o mesmo comprimento:  $\frac{1}{255}$ . Consequentemente, vemos que as faixas intervalares na conversão de  $\mathbb{E}$  para  $\mathbb{E}_{[0,1]}$  são distintas das que acabamos de ver (cf. (5.2)).

### Entradas Negativas

No modelo matemático não há explicitamente restrições de sinal sobre as componentes matriciais, enquanto que, no contexto prático, sim, há uma restrição implícita de que as componentes não apenas precisam ser não negativas mas, idealmente, deveriam estar no intervalo  $[0, 1]$ . Por isso, não é razoável esperar que os resultados numéricos voluntariamente se adequem às expectativas da aplicação.

A razão pela qual consideramos o valor absoluto das imagens – em lugar de um simples escalamento – é o fato de que o zero tem um papel central na adequação do problema prático ao modelo matemático, já que esse valor representa ausência de informação importante e está atrelado à esparsidade de uma das componentes. Optamos pelo valor absoluto por entender que, numa eventual ocorrência de valores negativos, a informação mais relevante está na distância ao zero. Destacamos que o descarte dos sinais em elementos da matriz só ocorre na etapa de exibição e não altera as soluções encontradas nem as medidas que se referem ao espaço de otimização.

Em experimentos preliminares com instâncias práticas de [21], observamos a tendência das soluções numéricas se restringirem ao intervalo  $[-1, 1]$  e, como veremos, se mostraram consistentes com nossos outros resultados. Isto talvez possa ser justificado pelo fato de que a envoltória convexa para a cardinalidade e o posto são construídas sobre a bola unitária, e não apenas sobre um conjunto de valores não negativos. Candès *et al.* não explicam como abordaram a provável presença de entradas negativas nem como procederam na interpretação desses valores ao exibir as respectivas imagens.

### 5.3 Soluções de Referência

Diferentemente do que ocorre com instâncias construídas com o propósito dos testes, como as artificiais aleatórias, as soluções das instâncias práticas podem sequer existir, e, mesmo que existam, é possível que não estejam disponíveis. No nosso caso não é diferente. É muito improvável que uma matriz oriunda de uma gravação real qualquer satisfaça as condições teóricas de recuperabilidade do Problema (2.1) – e, de fato, essa averiguação não está no escopo deste trabalho. A iniciativa do uso de vídeos de vigilância nos testes se justifica porque há indícios de que os resultados obtidos são satisfatórios, como mostra, por exemplo, o principal artigo no qual este trabalho se embasa [21]. Além disso, no caso prático, dispomos do crivo da aplicação para avaliar se uma solução é razoavelmente boa ou não, e podemos conferir visualmente o resultado.

Em algumas situações, os vídeos são disponibilizados em seus conjuntos de teste junto com o chamado *ground truth*, que é o plano de fundo correspondente a um único *frame* do vídeo, correspondente a uma coluna matricial. O *ground truth* pode ser uma foto tomada quando não há movimentação na cena ou, também, pode ser construído a partir de edições das imagens que compõem o vídeo. Usar o *ground truth* como solução significaria replicar essa imagem na mesma quantidade de *frames* do vídeo original e implicaria em supor posto unitário para a matriz de solução. Claramente, isto contradiria as nossas hipóteses que visam possibilitar a acomodação de pequenas variações no plano de fundo ao modelo, devidas, por exemplo, a mudanças de luminosidade (luzes ligadas ou desligadas, movimento do sol, etc.), a objetos que oscilam (plantas ao vento, ponteiros de relógio, etc.) ou, mesmo, a pequenas mudanças definitivas no plano de fundo (a retirada de um objeto, a abertura de uma janela, etc.). Portanto, na nossa situação, os *ground truths* não correspondem a uma solução do problema.

Sendo assim, optamos por construir soluções para os nossos problemas-testes de forma a poder medir o que chamaremos de erro, em analogia ao que fizemos para as instâncias aleatórias. A estratégia se baseia na confiabilidade do ADMM enquanto método que já foi – e de certa forma ainda é – considerado o estado da arte para esse tipo de problema. Para obter as soluções, faremos o ADMM ser executado por um número mínimo de 500 iterações até que sua infactibilidade relativa (medida de otimalidade) chegue a  $10^{-8}$  ou menos. Como o SPGM é um método que trabalha sob factibilidade, poderia não ser justo adotar como solução matrizes infactíveis. Portanto, a tática neste ponto é viabilizar com relação à matriz  $A$  cada componente do ADMM separadamente, e escolher em seguida aquela que conduzir a um menor valor para a função objetivo. A essas soluções chamaremos de soluções de referência.

**Definição 13. (Solução de Referência.)** Considere o Problema (2.7) com matriz de entrada  $A$  oriunda de uma gravação em vídeo. Seja  $(\hat{S}, \hat{L})$  a solução fornecida pelo ADMM após satisfazer o critério de parada que exige que o número de iterações ( $nIter$ ) esteja entre 500 e 20000 e que a infactibilidade relativa seja menor do que ou igual a  $10^{-8}$ , isto é,

$$500 \leq nIter \leq 20000 \quad \text{e} \quad \frac{\|A - \hat{S} - \hat{L}\|_F}{\|A\|_F} \leq 10^{-8}. \quad (5.9)$$

Chamaremos de *Solução de Referência* do problema o par  $(S^*, L^*)$  dado por

$$(S^*, L^*) = \begin{cases} (\hat{S}, A - \hat{S}), & \text{se } \Psi(\hat{S}, A - \hat{S}) < \Psi(A - \hat{L}, \hat{L}), \\ (A - \hat{L}, \hat{L}), & \text{caso contrário.} \end{cases} \quad (5.10)$$

Por simplicidade, podemos nos referir a  $(S^*, L^*)$ ,  $S^*$  e  $L^*$  apenas como “solução”, sem prejuízo da compreensão pelo contexto.

Note que, com infactibilidade tão baixa, o impacto da viabilização tende a ser mínimo, principalmente no Espaço de Exibição.

## 5.4 Medições Auxiliares

Vimos que o contexto da aplicação demanda soluções *suficientemente boas* a um custo indeterminado, que depende individualmente da situação do usuário, mas que se supõe *baixo*.

Agora que dispomos de um conceito de solução de referência, podemos instituir no Espaço de Otimização um erro relativo análogo ao da Definição 14 como uma forma de quantificar a qualidade dos candidatos a solução do problema de otimização. Pelo que discutimos anteriormente, essa qualidade da solução, em última instância, está intrinsecamente associada à sua correspondente no Espaço de Exibição. Portanto, é natural que definamos nesse espaço novas medições (ou medidas, mantido o sentido) que complementem o erro relativo na quantificação de qualidade.

Quanto ao custo, devido à indeterminação sobre o orçamento, é importante que as medições continuem sendo feitas ao longo das iterações, em termos do número de decomposições do tipo SVD, de forma que os resultados e as conclusões possam ser adaptados a diferentes disponibilidades de orçamento.

Tendo isso em mente, redefinimos a seguir o Erro Relativo (ER), no Espaço de Otimização, e duas novas medidas de erro, agora no Espaço de Exibição, chamadas de Píxeis Errados em Porcentagem (PEP) e Erro Médio por Píxel Errado (EMPE).

**Definição 14 (Erro Relativo (ER) nas Instâncias Práticas).** No caso das chamadas instâncias práticas, definimos como Erro Relativo aquele de acordo com a Definição 8, em que a matriz de referência é a solução de referência da Definição 13.

**Definição 15 (Medidas no Espaço de Exibição).** Considere duas matrizes  $X_I$  e  $X_I^*$  no Espaço de Exibição  $\mathbb{E} = \{0, 1, \dots, 255\}^{m \times n}$  e assumamos que  $X_I$  é uma tentativa de aproximar a solução de referência  $X_I^*$ . A fim de compará-las, definimos as seguintes medidas:

- (a) *Píxeis Errados em Porcentagem* (PEP): razão entre a quantidade de píxeis errados e a dimensão do espaço ao qual as matrizes pertencem, expressa na forma percentual

$$\text{PEP}(X_I; X_I^*) = 100 \cdot \frac{\text{nnz}(X_I - X_I^*)}{m \cdot n}. \quad (5.11)$$

- (b) *Erro Médio por Pixel Errado* (EMPE): o quanto, em média, cada pixel errado na matriz de aproximação difere em magnitude do pixel correspondente na solução de referência

$$\text{EMPE}(X_I; X_I^*) = \frac{\|\text{vec}(X_I - X_I^*)\|_1}{\text{nnz}(X_I - X_I^*)}, \quad (5.12)$$

em que assumimos adicionalmente que  $X_I \neq X_I^*$ .

Assim como na definição do Erro Relativo, os símbolos que representam as matrizes de aproximação e de referência poderão ser omitidos nos argumentos quando estas forem identificáveis pelo contexto, de forma que tanto os nomes das medidas quanto seus valores poderão ser representados pelas respectivas siglas. Quando necessário, incorporaremos os rótulos “-A”, “-S”, “-L” ou “-SL” às siglas para indicar a qual matriz os dados se referem: respectivamente, à matriz A de entrada, à componente S da esparsidade, à componente L do posto ou, ainda, à matriz [S L] resultante da justaposição de ambas componentes.

## 5.5 Precisão-Alvo

Até o momento, conseguimos elaborar um conceito de solução para as instâncias do problema prático e pudemos identificar que a demanda por precisão está mais dirigida ao que denominamos Espaço de Exibição do que propriamente ao Espaço de Otimização. Com isso, e também com as ferramentas detalhadas neste capítulo, podemos agora conjecturar um valor para o Erro Relativo que seja suficiente para obter bons resultados de compromisso entre a qualidade de exibição das imagens e o esforço computacional para obtê-las. Além disso, com as medidas auxiliares que definimos há pouco, poderemos avaliar durante os experimentos computacionais a legitimidade desse nível de precisão que estamos por estabelecer.

A pergunta cuja resposta nos propomos a discutir é a seguinte: qual é o nível de precisão que um método computacional precisa atingir para obter uma imagem

satisfatória na decomposição posto-esparsidade com vídeos de vigilância? Obviamente, quanto mais precisão se almeja, maior tende a ser o esforço computacional necessário. Também é evidente que, se o Erro Relativo alcançado for da ordem de, por exemplo,  $10^{-10}$ , é *provável*<sup>1</sup> que a imagem que o originou no Espaço de Exibição seja idêntica à solução de referência, dada a natureza combinatória das matrizes de 8 bits, como vimos antes. Igualmente provável é o gasto exorbitante necessário para atingir tal precisão – isso supondo que seja possível atingi-la.

A ideia aqui é, portanto, conjecturar para o Erro Relativo uma tolerância de compromisso entre o orçamento e a qualidade da imagem resultante. No que segue, apresentamos algumas justificativas que sugerem que a escolha do valor 0.1% para a tolerância com o Erro Relativo é suficiente para obter resultados adequados ao propósito da aplicação. Esse limitante norteará os experimentos computacionais discutidos no próximo capítulo.

Na seção anterior, vimos que, ao converter uma matriz  $\widehat{X} = (x_{ij})$  do intervalo  $[0, 1]$  para o Espaço de Exibição, particionamos o intervalo unitário em 256 subintervalos e impomos uma correspondência entre cada um deles e os valores inteiros de 0 a 255, de acordo com suas magnitudes. Vimos também que a conversão é feita individualmente a nível de píxel. *Grosso modo*, o comprimento padrão de cada subintervalo<sup>2</sup> é igual a  $\frac{1}{255}$ .

Considere que a matriz  $\widehat{X}^* = (x_{ij}^*)$  seja a versão escalada para  $[0, 1]^{m \times n}$  da solução  $X^*$ . Suponhamos por um momento que o erro relativo em cada entrada de  $\widehat{X}$  não ultrapasse 0.1%, isto é, que

$$|x_{ij} - x_{ij}^*| \leq \frac{0.1}{100} \cdot |x_{ij}^*|. \quad (5.13)$$

Disto segue diretamente que o próprio erro absoluto também pode ser limitado, já que as entradas estão restritas a  $[0, 1]$ :

$$|x_{ij} - x_{ij}^*| \leq 10^{-3}. \quad (5.14)$$

Comparemos então essa margem de erro por componente com os subintervalos de comprimento  $\frac{1}{255}$  que definirão os valores inteiros dos píxeis na conversão para o Espaço de Exibição. Se uma entrada  $x_{ij}$  qualquer estiver localizada na faixa central que ocupa metade do subintervalo ao qual pertence, a segurança da margem de erro garante que essa entrada será convertida ao número inteiro correto. De fato, a margem

<sup>1</sup> O motivo de ser apenas provável, e não certo, é que o erro relativo mede distância entre a aproximação e a solução, mas não impede que a conversão para números inteiros seja exata. Basta pensar em uma dimensão: os dois pontos podem estar muito próximos porém, não sendo vizinhos imediatos na representação em ponto flutuante, pode haver entre eles durante a conversão um limiar de classificação.

<sup>2</sup> A caracterização dos subintervalos está na Observação da Subseção 5.2 na página 114.

de erro absoluto de 0.001 para mais ou para menos corresponde a uma faixa de 0.002 centrada em  $x_{ij}$ , e isto num subintervalo de tamanho  $\frac{1}{255}$  representa  $0.002 \div \frac{1}{255} = 51\%$  de seu comprimento. Ocorre que, num caso geral,  $x$  pode estar tão próximo à fronteira do subintervalo quanto a representação em ponto flutuante permite – o que pode chegar a 17 algarismos significativos, um valor que para todos os efeitos é zero. Ainda assim, com a segurança que a hipótese sobre o erro nos dá, teremos a plena certeza de que o píxel desviará do valor correto em *no máximo* um tom de cinza. Esse desvio de um tom na escala de cor, como vimos na Seção 5.2, é tão baixo que dificilmente o olho humano típico será capaz de percebê-lo.

É importante observar que o raciocínio que acabamos de desenvolver se baseia em duas hipóteses. A primeira delas é a de que tanto a matriz de aproximação quanto a da solução foram escaladas para o intervalo  $[0, 1]$ , algo que de fato não ocorre quando calculamos o Erro Relativo (8) no decorrer da execução dos algoritmos. O que definimos como Solução de Referência está relacionado com a solução (no Espaço de Otimização) que o ADMM pode fornecer, sob determinado orçamento, na resolução do Problema (2.5), e seria portanto um alvo a ser atingido por outros métodos computacionais que contemplassem esse problema. A segunda hipótese trata do erro relativo por entrada matricial, algo que o Erro Relativo (8), definido na norma de Frobenius, não é capaz de medir diretamente, embora haja uma relação entre as medidas.

À luz desses aspectos, consideramos que o Erro Relativo de 0.1%, ou seja,  $10^{-3}$ , constitui uma precisão suficiente para obter imagens que desviem da solução em poucos tons de cinza por píxel no Espaço de Exibição. Essa será a precisão-alvo que, neste trabalho, classificará como resolvido o problema da separação dos vídeos de vigilância. Com a perspectiva das Medidas Auxiliares da Seção 5.4, nossos experimentos mostrarão empiricamente que, de fato, esse nível de precisão conduz a resultados condizentes com o previsto e que são, portanto, bastante satisfatórios.

## 6 INSTÂNCIAS PRÁTICAS DE TESTE: IMAGENS DE VIDEOVIGILÂNCIA

Com a finalidade de estudar e comparar os desempenhos do Método dos Multiplicadores com Direções Alternadas (ADMM) e do Método do Gradiente Proximal Espectral (SPGM), selecionamos 28 vídeos de situações reais advindos de dois conjuntos de teste disponibilizados publicamente pela *Internet* nas páginas de seus respectivos projetos de pesquisa.

### 6.1 Problemas-testes de Videovigilância

O primeiro conjunto de vídeos é composto por uma seleção de 14 das 54 cenas do Projeto CAVIAR<sup>1</sup> [23], escolhidas por nós de forma a contemplar uma variabilidade de situações que pudesse ser representativa para nosso estudo, incluindo ocorrência de grandes movimentações, abandono de objeto, conflito físico entre pessoas, entrada e saída de transeuntes na cena, etc. Já o segundo conjunto, também com 14 vídeos, coincide com o Conjunto de Dados SBI [12], uma iniciativa que visa padronizar os testes de avaliação comparativa na área de *Background Initialization*<sup>2</sup>. Inicialmente, este trabalho contava apenas com os vídeos do primeiro grupo, sendo que os do segundo grupo foram incluídos já nas etapas finais, quando tomamos conhecimento de sua existência.

#### Projeto CAVIAR

O Projeto CAVIAR (*Context Aware Vision using Image-based Active Recognition*<sup>3</sup>) foi um projeto sobre reconhecimento de imagens liderado pelo Professor Robert B. Fisher da Escola de Informática da Universidade de Edimburgo, na Escócia, e financiado pela Sociedade da Informação da Comissão Europeia. Sua base de dados é separada em dois subconjuntos de acordo com as características das filmagens: o primeiro deles compreende 28 cenas feitas no pátio de entrada de um laboratório francês, enquanto o segundo traz 26 cenas de um centro comercial português, gravadas com duas tomadas diferentes de um mesmo corredor. Tratam-se, portanto, de três perspectivas de gravação, cada uma delas registrada com lente grande angular e câmera fixada.

<sup>1</sup> EC Funded CAVIAR project/IST 2001 37540, <https://homepages.inf.ed.ac.uk/rbf/CAVIAR/> (último acesso em 10/10/2021).

<sup>2</sup> Inicialização de Plano de Fundo, em tradução livre.

<sup>3</sup> Visão Sensível ao Contexto usando Reconhecimento Ativo baseado em Imagens, em tradução livre.

Cada um dos 14 vídeos que selecionamos do projeto escocês é identificado neste trabalho por um número formado por dois dígitos,  $\underline{d_1 d_2}$ , sendo o primeiro associado a uma dentre as três perspectivas de gravação e o segundo tendo o papel de contador. Assim, os vídeos  $\underline{1 d_2}$  e  $\underline{2 d_2}$  refletem, respectivamente, as visões transversal e longitudinal do corredor do centro comercial, enquanto os  $\underline{3 d_2}$  correspondem ao pátio de entrada do laboratório. A Tabela 8 exhibe as identificações e breves descrições para esses vídeos, enquanto a Tabela 9 fornece a seleção de *frames* que fizemos a partir dos arquivos originais.

Os arquivos de vídeo são originalmente disponibilizados no formato MPG com resolução de  $384 \times 288$  píxeis por quadro e taxa de reprodução de 25 quadros por segundo. Alternativamente, o *site* fornece também os arquivos JPG relativos aos vídeos. Originalmente a cores (RGB), convertemos os vídeos para escala de cinza conforme o procedimento descrito na Equação (E.6) do Apêndice E. Chamaremos esse conjunto de problemas de **Grupo CAVIAR**.

Tabela 8 – Breve descrição das cenas das instâncias do grupo CAVIAR, uma seleção do projeto de mesmo nome.

Cód.	Nome	Breve descrição
11	EnterExitCrossingPaths1front	Duas pessoas se cruzam na entrada de uma loja e um casal caminha no corredor.
12	WalkByShop1front	Casal passeando pelo corredor, pessoas entrando e saindo das lojas.
13	TwoLeaveShop1front	Casal passeia após sair de uma loja.
14	OneStopMoveEnter1front	Pessoa para em frente a uma loja, e em seguida entra e sai do estabelecimento. Há cinco grupos de pessoas andando pelo corredor.
21	OneStopMoveEnter1cor	Pessoa para em frente a uma loja, e, em seguida, entra e sai do estabelecimento. Há cinco grupos de pessoas andando pelo corredor.
22	WalkByShop1cor	Casal passeando pelo corredor, pessoas entrando e saindo das lojas.
23	OneLeaveShopReenter2cor	Pessoa sai da loja para depois voltar a entrar nela. Quatro pessoas caminham pelo corredor.
24	TwoLeaveShop1cor	Casal passeia após sair de uma loja.
25	OneShopOneWait2cor	Pessoa para em frente a uma loja, e, em seguida, entra e sai do estabelecimento. Vários grupos de pessoas andam pelo corredor.
31	Meet_Crowd	Grupo de 4 pessoas se encontra, caminha e se separa.
32	Browse_WhileWaiting1	Pessoa perambulando e fazendo pequenas pausas eventuais
33	LeftBox	Pessoa carregando uma caixa na mão entra em cena, a deposita no pátio e sai.
34	Fight_OneManDown	Duas pessoas se encontram, brigam, uma delas cai e o outro foge apressadamente.
35	Walk1	Pessoa andando em linha reta

## Conjunto de Dados SBI

O Conjunto de Dados SBI (*Scene Background Initialization*<sup>4</sup>) [88] é uma iniciativa dos pesquisadores italianos Lucia Maddalena e Alfredo Petrosino, que visa contribuir com a padronização das instâncias de teste na avaliação comparativa de algoritmos de inicialização de plano de fundo [67]. A proposição das 14 instâncias que

<sup>4</sup> Inicialização de Fundo de Cena, em tradução livre.

Tabela 9 – Características dos recortes dos vídeos das instâncias do Grupo CAVIAR com relação às versões originalmente disponibilizadas. Em todos os casos, as dimensões dos *frames* são  $384 \times 288$ , assim como nas versões originais.

Cód.	Nome	Frames originais	Frames utilizados	Qtd. de frames
11	EnterExitCrossingPaths1front	1–383	1–383	383
12	WalkByShop1front	1–2360	1603–2352	750
13	TwoLeaveShop1front	1–1343	1–450	450
14	OneStopMoveEnter1front	1–1590	251–1000	750
21	OneStopMoveEnter1cor	1–1590	1–750	750
22	WalkByShop1cor	1–2360	1153–1902	750
23	OneLeaveShopReenter2cor	1–560	1–560	560
24	TwoLeaveShop1cor	1–1343	1–750	750
25	OneShopOneWait2cor	1–1462	1–750	750
31	Meet_Crowd	1–483	1–483	483
32	Browse_WhileWaiting1	1–792	7–756	750
33	LeftBox	1–864	105–854	750
34	Fight_OneManDown	1–960	8–457	450
35	Walk1	1–612	7–612	606

compõem a seleção de vídeos foi feita inicialmente em um Workshop em 2015 ([67]), ainda com apenas 7 vídeos, e foi ampliada posteriormente para o número de 14 vídeos, sendo então difundida mediante uma publicação em parceria com o pesquisador francês Thierry Bouwmans ([12]).

As sequências de imagens do Conjunto de Dados SBI provieram de outros oito conjuntos de dados preexistentes: ATON, COST211, PBI, ITEA CANDELA, CAVIAR, IBM, MPI Informatik e RPI ISL (mais informações em [12, Seção 4]). A idealização dessa nova base de dados considerou situações variadas que contemplassem as dificuldades que normalmente se apresentam em seu contexto de aplicação, como por exemplo cenas movimentadas, elementos oscilatórios, pequenas tremulações na câmera etc. Além de selecionar os vídeos, os autores também fizeram algumas modificações nos arquivos originais, como a eliminação de quadros e a alteração de certas resoluções de imagem. As Tabelas 10 e 11 contêm essas informações de corte, resolução e origem dos vídeos, além de providenciarem uma breve descrição de cada cena. Assim como no caso do grupo CAVIAR, convertemos os vídeos de RGB para escala de cinza seguindo a Equação (E.6) do Apêndice E. A esse conjunto de problemas chamaremos de **Grupo SBI**.

### Ground Truths

Cada sequência de imagens dos grupos CAVIAR e SBI vem acompanhada do chamado *ground truth* – imagem de referência, em tradução e adaptação livres – para o plano de fundo do vídeo. De acordo com seus autores, o *ground truth* consiste em uma única imagem que retrata apenas o plano de fundo de um instante da cena e foi obtida de diferentes formas, dependendo da instância. Em alguns casos, foi selecionado um quadro do vídeo original no qual os objetos do primeiro plano não

aparecem, sendo esse quadro removido posteriormente da sequência de imagens do banco de dados. Outra maneira de obter o *ground truth* foi mediante a junção de recortes de diferentes quadros, ou mesmo calculando-se uma mediana temporal por regiões do plano de fundo. As Figuras 11 e 12 exibem as imagens dos *ground truths* disponibilizados.

Apesar de estarem disponíveis, os *ground truths* não serão utilizados neste trabalho como referências por razões que já comentamos na Seção 5.3 e reforçamos agora. Essa imagem de referência se trata de uma única foto por cenário e precisaria ser replicada para corresponder a um vídeo contendo apenas tal plano de fundo. Proceder dessa forma comprometeria um aspecto relevante do processo de separação, da maneira que é conduzido aqui: estaríamos desconsiderando ou mesmo penalizando a capacidade da estratégia adotada em acomodar pequenas alterações no próprio plano de fundo, como por exemplo variações de luminosidade no ambiente ou abandono de objetos em certo instante, no decorrer da cena.

Conforme adiantamos na Seção 5.3, nossa solução de referência virá não do *ground truth* mas de uma execução de longo prazo ou baixa infactibilidade do próprio ADMM, seguida de uma viabilização que, na prática, faz pouca diferença. Numa comparação entre dois algoritmos, permitir que o oponente estabeleça a solução de referência torna a avaliação de desempenho ainda mais desafiadora para o opositor. Além disso, procedendo dessa forma, focamos a comparação nos métodos computacionais e não tanto na estratégia de decomposição em si, que pode inclusive não funcionar para alguns vídeos.

Tabela 10 – Breve descrição das cenas das instâncias dos vídeos do grupo SBI. Traduzida e adaptada da tabela disponível na página do projeto ([88]).

Nome	Breve descrição
Board	Homem se movendo na frente de um painel, com sombras suaves.
Candela_m1.10	Homem entrando e saindo de uma sala, abandonando uma bolsa na maioria dos quadros.
CAVIAR1	Pessoas caminhando lentamente ao longo de um corredor, com sombras suaves.
CAVIAR2	Pessoas entrando e saindo de uma loja, parando apenas por alguns quadros.
CaVignal	Homem de pé na maioria dos quadros e depois se movendo.
Foliage	Carros estacionados obstruídos por grandes folhas ondulantes.
HallAndMonitor	Pessoa ambulante e bolsa abandonada na mesma região da imagem para a maioria dos quadros.
HighwayI	Movimento rápido de carros ao longo de uma rodovia, com sombras fortes e pequenas oscilações de câmera.
HighwayII	Movimento rápido de carros ao longo de uma rodovia, com sombras fortes e pequenas oscilações de câmera.
HumanBody2	Pessoas caminhando rapidamente dentro de casa, com sombras suaves.
IBMtest2	Pessoas caminhando rapidamente pelos corredores internos.
PeopleAndFoliage	Carros estacionados obstruídos por pessoas em movimento e grandes folhas ondulantes.
Snellen	Gráfico de Snellen estacionário ocluído e sombreado por grandes folhas ondulantes.
Toscana	Muito poucas fotos de pedestres ao ar livre tiradas em momentos diferentes.

Tabela 11 – Instâncias do Grupo SBI. Dados extraídos integralmente da tabela disponível na página do projeto ([88]).

Nome	Proveniência dos Dados	Frames originais	Frames utilizados	Qtd. de frames	Resolução original	Resolução final
Board	PBI	0–227	0–227	228	200 × 164	200 × 164
Candela_m1.10	Candela	0–855	85–435	351	352 × 288	352 × 288
CAVIAR1	CAVIAR	0–725	115–724	610	384 × 288	384 × 256
CAVIAR2	CAVIAR	0–1500	900–1360	461	384 × 288	384 × 256
CaVignal	PBI	0–257	0–257	258	200 × 136	200 × 136
Foliage	PBI	0–399	6–399	394	200 × 148	200 × 144
HallAndMonitor	COST 211	0–299	4–299	295	352 × 240	352 × 240
HighwayI	ATON	0–439	0–439	440	320 × 240	320 × 240
HighwayII	ATON	0–499	0–499	500	320 × 240	320 × 240
HumanBody2	RPI ISL	0–898	70–810	741	320 × 240	320 × 240
IBMtest2	IBM	0–1750	1027–1117	91	320 × 240	320 × 240
PeopleAndFoliage	PBI	0–349	0–340	341	320 × 240	320 × 240
Snellen	PBI	0–333	0–320	321	146 × 150	144 × 144
Toscana	MPI Informatik	0–5	0–5	6	2272 × 1704	800 × 600

## 6.2 Soluções de Referência para as Instâncias de Videovigilância

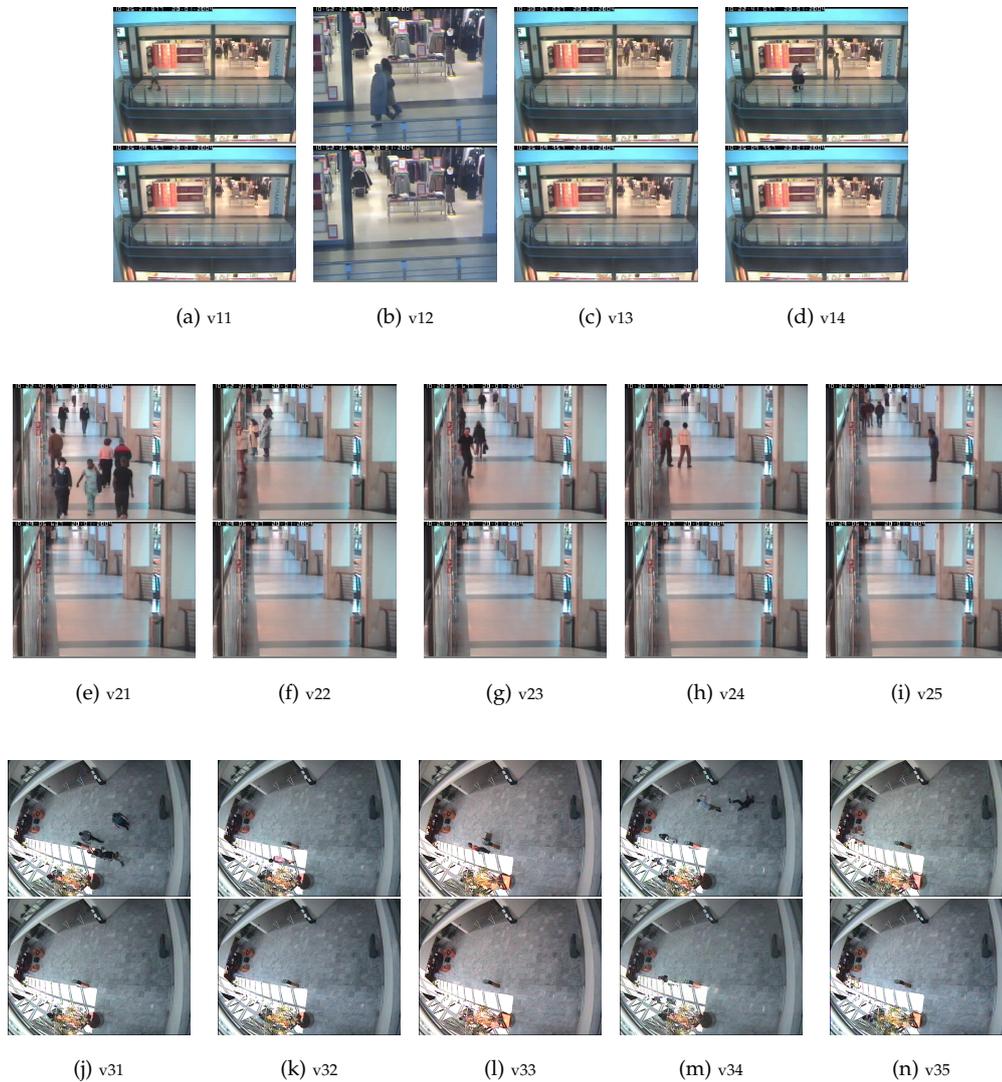
Antes de exibir os resultados dos experimentos para obtenção das soluções de referência, retomemos alguns aspectos discutidos nos capítulos anteriores. Diferentemente do que ocorre com as instâncias aleatórias, para as quais temos não só a garantia teórica da separabilidade como também a própria solução do problema, com as instâncias de videovigilância pode ser impossível realizar a decomposição matricial almejada por meio da formulação convexa que adotamos. Mesmo nos casos em que a decomposição é possível, não dispomos da solução de antemão. No entanto, é de bastante proveito estabelecer pontos referenciais com relação aos quais podemos calcular “erros”, de acordo com as Definições 14 e 15, e assim poder medir de diferentes formas a qualidade dos iterandos obtidos pelos métodos em análise.

Vimos que uma maneira de contornar o aspecto da incerteza sobre a separabilidade e, ao mesmo tempo, obter uma medida de erro relativo é considerar como **Solução de Referência** (cf. Definição 13) o ponto ótimo viabilizado proveniente de uma execução apurada do algoritmo estado-da-arte para essa classe de problemas, o ADMM. Note que, seguindo esse procedimento, nos casos em que a separação não seja possível, obteremos como solução um iterando que ao menos é atingível com a formulação convexa do problema. Assim, as análises de erro relativo devem ser lidas com a consciência de que se mede a distância relativa a um dos melhores pontos que se pode obter com o ADMM.

### Experimentos de obtenção da Solução de Referência

As soluções de referência para as instâncias de videovigilância foram obtidas a partir da resolução do problema convexo original (2.5) por meio da execução do ADMM com número de iterações não menor do que 500 e limitado a 20000, além de

Figura 11 – *Frames* do vídeo original (linha superior) e do *ground-truth* (linha inferior) fornecido em conjunto com as matrizes de entrada das instâncias do Grupo CAVIAR.



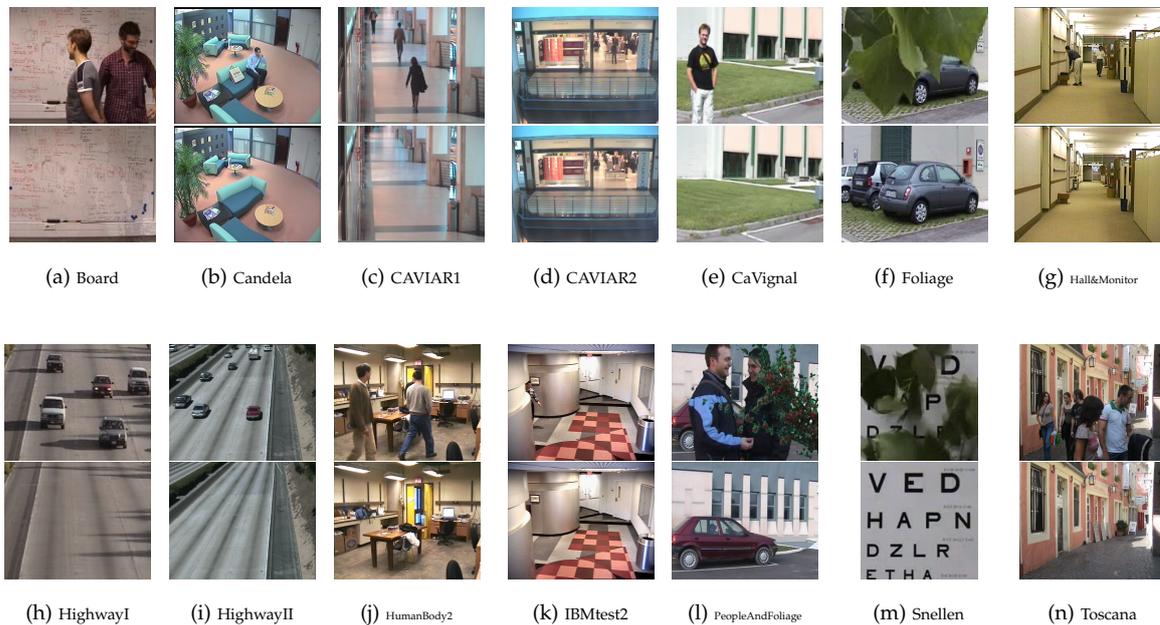
infactibilidade relativa menor do que ou igual a  $10^{-8}$ , como vimos na Equação (5.9). A Tabela 12 contém os valores dos principais parâmetros adotados nesses experimentos. Os resultados numéricos se encontram nas Tabelas 13 e 14. As Figuras 13 e 14 mostram um quadro de cada vídeo junto com a separação alcançada após a otimização.

Tabela 12 – Principais parâmetros do ADMM adotados nos experimentos de obtenção das Soluções de Referência. As dimensões  $n$  de cada instância estão exibidas nas Tabelas 13 e 14.

$\rho$	$\epsilon_{\text{relinf}}$	$\epsilon_x$	$m_{it}$	$M_{it}$
$\frac{1}{4} \frac{n^2}{\ \text{vec}(A)\ _1}$	$10^{-8}$	–Inf	500	20000

A configuração adotada se baseou em experimentos preliminares monitorando a variação de pixels no decorrer das iterações. O número mínimo de iterações

Figura 12 – *Frames* do vídeo original (linha superior) e do *ground-truth* (linha inferior) fornecido em conjunto com as matrizes de entrada das instâncias do Grupo SBI.



foi imposto devido à existência de dois casos excepcionais: em duas das 28 instâncias de videovigilância (CAVIAR1 e CAVIAR2), o ADMM atingiu o patamar mínimo de infactibilidade logo nas iterações iniciais, sem no entanto realizar a separação desejada para o plano de fundo. Tendo em mente que não há garantias teóricas de recuperabilidade para as instâncias práticas, optamos por estipular o número mínimo de 500 iterações como critério adicional, que na prática só faz diferença para esses dois casos, como forma de avaliar se algum êxito poderia ser obtido. As demais configurações de execução seguem o padrão adotado neste trabalho para o método.

No caso da instância CAVIAR2, após as 500 iterações obtivemos um bom resultado visual de separação. Por outro lado, com a instância CAVIAR1, a qualidade aparente da separação não foi satisfatória, possivelmente por uma dificuldade intrínseca à dinâmica da cena ou mesmo a um valor desfavorável do parâmetro  $t$ . Ainda assim, em ambos os casos, os pontos obtidos foram considerados como solução de referência por serem o melhor que se pode obter com o estado-da-arte de primeira ordem. Curiosamente, a decomposição alcançada com o SPGM se mostrou mais eficiente na separação do que a solução de referência obtida a partir do ADMM. Já com as instâncias CaVignal e Candela\_m1.10, o ADMM não obteve a factibilidade desejada e esgotou o orçamento de 20 mil iterações. Como constatamos pelas Tabelas 13 e 14, essas foram as duas únicas instâncias em que isso ocorreu. Com exceção das quatro instâncias mencionadas, com todas as outras, a parada do ADMM foi por factibilidade.

Vale lembrar que, apesar da baixa infactibilidade, entendemos que não faria sentido manter como solução um ponto que, a rigor, é infactível. Por isso, em cada

instância, contornamos a situação escolhendo dentre as duas matrizes-componente aquela que conduziisse a um menor valor da função objetivo correspondente ao problema irrestrito:  $\Psi(S, A - S)$  ou  $\Psi(A - L, L)$  – veja a Equação (4.3). Como esperado, esse processo de viabilização teve baixo impacto visual e em cada instância alterou não mais do que poucas dezenas de píxeis no Espaço de Exibição.

Figura 13 – Grupo CAVIAR. *Frames* do vídeo original (linha superior), da componente do posto (linha intermediária) e da componente da esparsidade (linha inferior) da Solução de Referência.

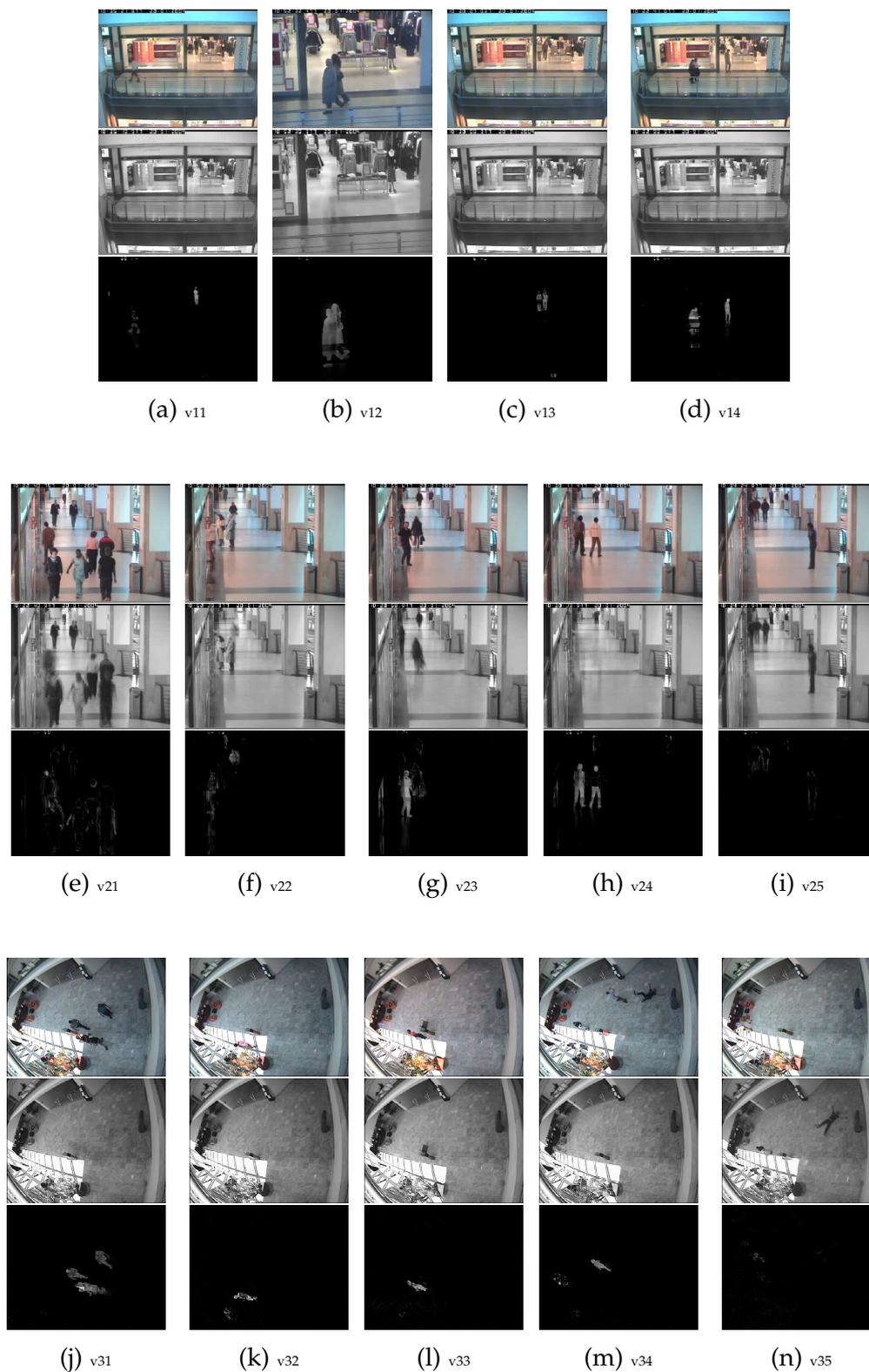


Figura 14 – Grupo SBI. *Frame* do vídeo original (linha superior), da componente de posto (linha intermediária) e da componente de esparsidade (linha inferior) da Solução de Referência.

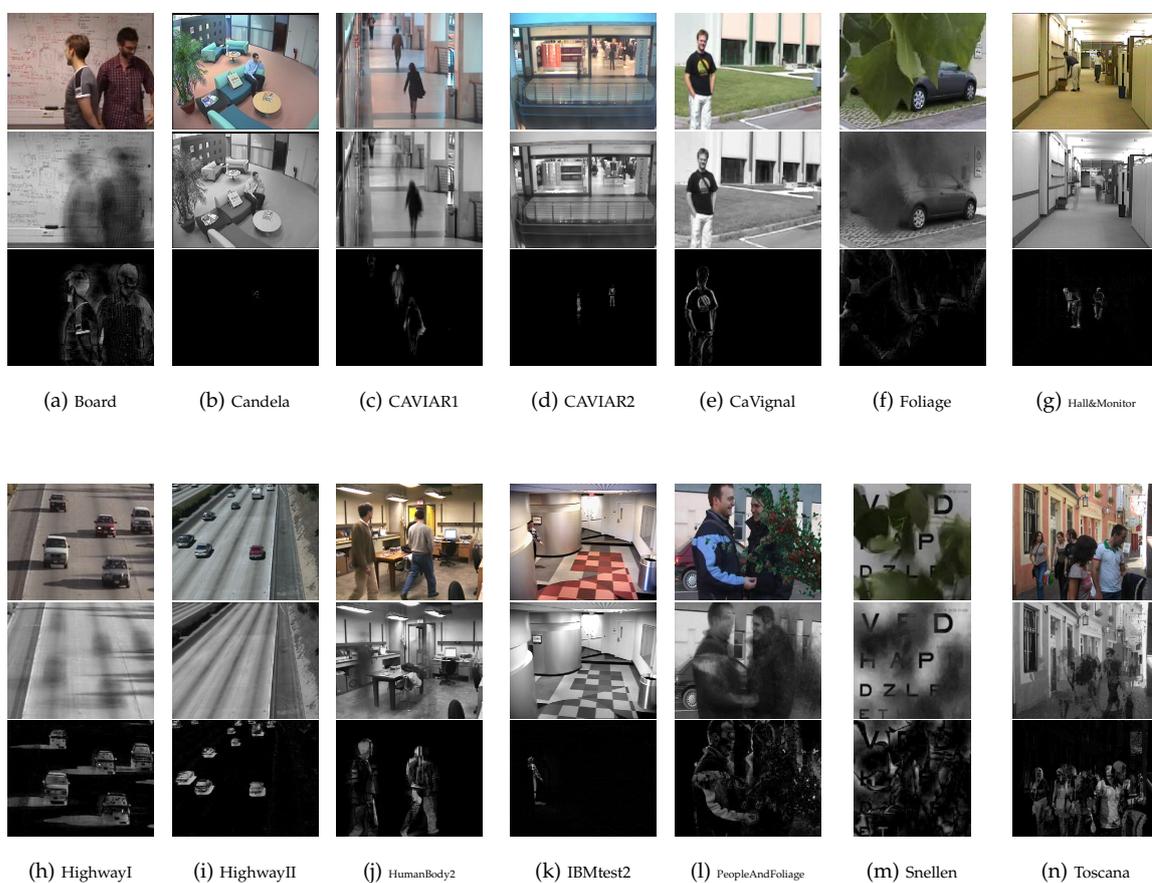


Tabela 13 – Resultados dos experimentos de obtenção das Soluções de Referência para o Grupo CAVIAR. O parâmetro  $t$  é comum a todas as instâncias e tem valor aproximado de 0.003 ou, ainda, 0.3%. Em todas as instâncias, a parada se deu por factibilidade.

Vídeo	m	n	nIter	objval	spr	≈spr	rnk	≈rnk	tempo (s)	CPU (s)
11	110592	383	4272	4478.64204	68	64	184	182	2.4e+04	1.0e+05
12	110592	750	8621	8073.95840	70	65	372	367	1.1e+05	4.9e+05
13	110592	450	6093	5025.52008	68	64	219	215	4.1e+04	1.8e+05
14	110592	750	6330	7499.00420	67	63	369	365	8.2e+04	3.6e+05
21	110592	750	6424	13127.99467	73	69	370	370	8.3e+04	3.7e+05
22	110592	750	5401	9277.03046	72	68	378	377	7.0e+04	3.1e+05
23	110592	560	8933	7241.54843	73	68	282	278	7.8e+04	3.4e+05
24	110592	750	8795	8351.19394	75	70	382	381	1.1e+05	5.0e+05
25	110592	750	8641	8974.96446	73	69	381	379	1.1e+05	4.9e+05
31	110592	492	9414	5831.48177	82	78	233	232	7.1e+04	3.1e+05
32	110592	750	9324	7377.22220	82	78	370	368	1.2e+05	5.4e+05
33	110592	750	7545	7854.67972	81	78	352	349	9.9e+04	4.4e+05
34	110592	959	7289	9242.89235	83	79	517	517	1.3e+05	5.8e+05
35	110592	606	9360	6803.28508	81	78	285	283	8.2e+04	7.1e+05

Tabela 14 – Resultados dos experimentos de obtenção das Soluções de Referência para o Grupo SBI.

Vídeo	m	n	t (%)	nIter	objval	spr	≈spr	rnk	≈rnk	tempo (s)	CPU (s)	parada
Board	32800	228	0.55	7002	4224.45868	77	75	119	119	5.3e+03	4.7e+04	feas
Candela_m1.10	101376	350	0.31	20000	4130.90784	72	58	196	192	7.6e+04	7.2e+05	iter
CAVIAR1	98304	610	0.32	2	101907.18713	100	100	1	1	2.2e+01	2.4e+02	feas
CAVIAR2	98304	460	0.32	2	71057.26604	100	100	1	1	1.3e+01	1.5e+02	feas
CaVignal	27200	258	0.60	20000	2650.45483	34	25	72	69	1.4e+04	5.5e+04	iter
Foliage	28800	394	0.59	2461	6217.06860	78	78	210	209	3.0e+03	1.3e+04	feas
HallAndMonitor	84480	296	0.34	6632	4690.19470	89	87	118	117	1.8e+04	7.4e+04	feas
Highway1	76800	440	0.36	796	8150.35570	85	84	217	217	3.0e+03	1.3e+04	feas
HighwayII	76800	500	0.36	840	6236.32175	94	93	289	289	3.7e+03	1.6e+04	feas
HumanBody2	76800	740	0.36	2384	10593.68647	83	81	353	353	1.8e+04	7.8e+04	feas
IBMtest2	76800	90	0.36	7709	2327.10551	91	90	25	25	4.8e+03	1.8e+04	feas
PeopleAndFoliage	76800	341	0.36	2352	9814.92408	81	80	175	175	6.7e+03	2.9e+04	feas
Snellen	20736	321	0.69	15331	3701.66763	76	75	174	173	1.2e+04	4.9e+04	feas
Toscana	480000	6	0.14	3622	1083.75987	79	79	3	3	8.0e+02	3.3e+03	feas

## 6.3 Análise Estrutural das Matrizes dos Vídeos

É comum em trabalhos que envolvem o Problema SLRMD que o foco dos experimentos computacionais esteja colocado sobre as instâncias de natureza aleatória e que, a partir de seus resultados, ocorram validações de parâmetros e tomadas de decisões que, em algum sentido, poderiam se estender a casos práticos [21, 103]. Se, por um lado, essas instâncias artificiais podem trazer as facilidades da adaptação às hipóteses da teoria de recuperabilidade e também da manipulação computacional, por outro, como veremos à continuação, elas podem estar distantes do cenário real de aplicação.

Algo que nem sempre é desenvolvido nos trabalhos publicados é o questionamento sobre quão representativas as matrizes aleatórias são, em particular, com relação às instâncias de gravação. A relevância dessa análise se dá na medida em que muitas vezes a comparação de desempenho de métodos computacionais é baseada nos resultados com as instâncias artificiais. O que fazemos a seguir é analisar a estrutura das matrizes dos vídeos e compará-las com as das instâncias aleatórias, ainda sob as perspectivas das entradas matriciais e também dos valores singulares.

### O que comparar

Assim como fizemos com as instâncias aleatórias na Seção 4.4, vamos analisar as distribuições das entradas e dos valores singulares de cada matriz envolvida,  $S$ ,  $L$  e  $A$ , usando *boxplots* e gráficos de distribuição para ilustrar os dados. A principal diferença, aqui, é que as componentes  $S$  e  $L$  são as soluções de referência obtidas por meio do ADMM em uma execução de longo prazo ou baixa infactibilidade, seguida de viabilização, como visto anteriormente neste capítulo e na Seção 5.3. Note que, na prática,  $S$  não é necessariamente esparsa, nem  $L$  tem indispensavelmente posto baixo, já que não estamos sob as hipóteses teóricas que garantem a separação a partir da resolução do problema de otimização. Essas matrizes sequer precisam ter seus elementos restritos a algum intervalo numérico predeterminado. Por outro lado, a matriz  $A$ , esta sim, garantidamente terá suas entradas limitadas ao intervalo  $[0, 1]$ , já que advém diretamente da conversão da imagem do Espaço de Exibição para o de Otimização.

#### 6.3.1 Entradas Matriciais

A Figura 15 contém os *boxplots* das distribuições das entradas de cada uma das três matrizes para cada um dos dois grupos de vídeos. Vale ressaltar que, como vimos na Subseção 4.1, nesse tipo de gráfico, cada caixa azul compreende o intervalo interquartil, isto é, aquele que abrange a metade dos dados em torno da mediana, a

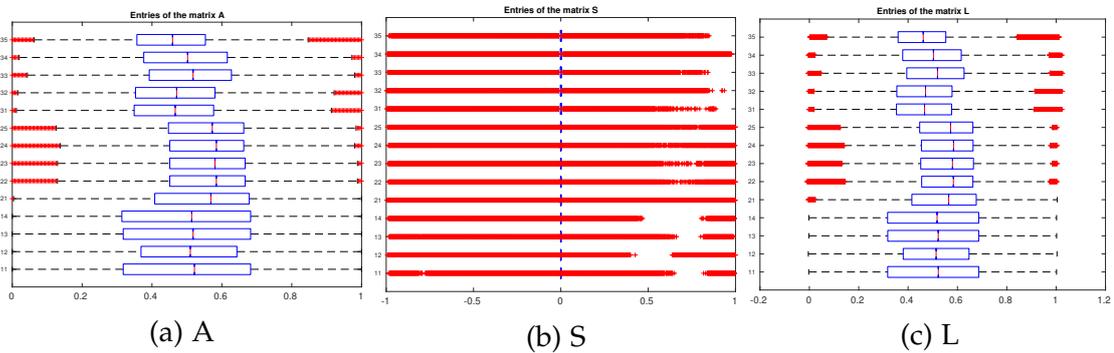
qual por sua vez é marcada como um segmento vertical vermelho. Os símbolos de dois segmentos cruzados (“+”), também em vermelho, denotam os valores extremos considerados discrepantes (*outliers*). Já a linha tracejada (*whisker*) é construída para se prolongar entre os valores classificados como “mínimo” e “máximo” do conjunto de dados, desconsiderados os *outliers*.

Iniciando pelas matrizes  $A$  do Grupo CAVIAR, percebemos pela Subfigura 15(a) que metade de suas entradas se concentra aproximadamente na faixa central contida em  $[0.3, 0.7]$ , com mediana em  $[0.4, 0.6]$ , e que o restante de suas entradas está disperso nas demais partes do intervalo unitário. Podemos observar que há uma certa simetria da distribuição em torno da mediana. Já no caso das matrizes  $A$  do Grupo SBI, na Subfigura 15(d), apesar das distribuições tipicamente ocuparem o intervalo unitário por completo, com menos ocorrência de instâncias com *outliers*, as medianas oscilam, entre os vídeos, para mais longe do centro do intervalo.

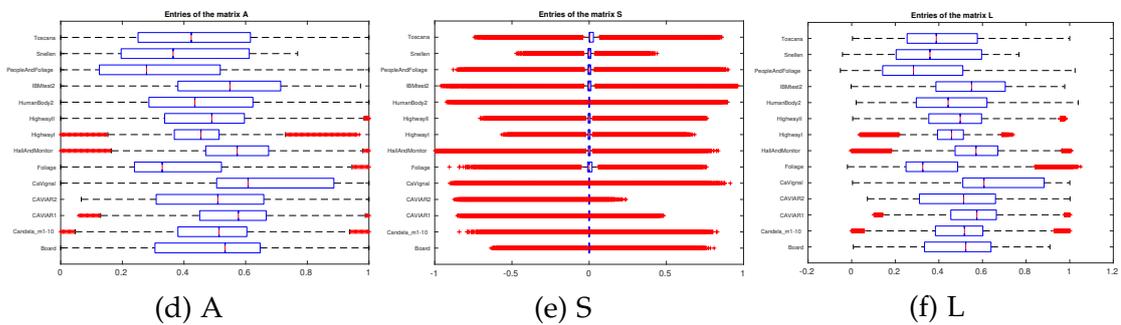
É possível destacar algumas interpretações em termos dos píxeis e de seus tons de cinza. Podemos afirmar, por exemplo, que nas gravações originais não há predominância nem de tons mais escuros (próximos a 0), como seria em ambientes mal iluminados, nem de tons mais claros (próximos a 1), como em situações com excesso de luminosidade. Enquanto no Grupo CAVIAR há predominância de tons de cinza intermediários na escala da cor, no Grupo SBI há instâncias em que os tons tendem a ser mais escuros, como em *PeopleAndFoliage*, e também aquelas cenas mais claras, como as de *CaVignal*.

Partindo para as componentes matriciais obtidas com o ADMM, vemos nas Subfiguras 15(c) e 15(f) que as distribuições das entradas de  $L$  são muito semelhantes às das respectivas matrizes  $A$ , inclusive na amplitude dos valores que as entradas ocupam. Isto não ocorre com os elementos de  $S$ , retratados nas Subfiguras 15(b) e 15(e), os quais estão espalhados por um intervalo aproximadamente igual a  $[-1, 1]$ , ainda que sua maioria esteja relativamente próxima à origem, como indica o tamanho diminuto da caixa azul localizada na vizinhança do zero. Quanto ao número de entradas não nulas em  $S$ , que a rigor é bastante alto, esse deve ser olhado com a consciência de que, tanto no sentido numérico quanto no sentido da aplicação, muitos elementos devem ser entendidos como nulos. As Tabelas 13 e 14 trazem as informações de esparsidade de  $S$ , exata e aproximada (com tolerância  $10^{-3}$ ), nas colunas  $\text{spr}$  e  $\approx\text{spr}$ , respectivamente. Podemos ver, então, que a componente  $S$  dos vídeos, na prática, não é muito esparsa. Além disso, podemos perceber pelos gráficos de distribuição de  $S$  uma certa tendência à simetria em torno da origem, na maioria das instâncias.

Figura 15 – Entradas das Matrizes dos Vídeos  
 Primeiro Grupo de Vídeos: CAVIAR



Segundo Grupo de Vídeos: SBI



Entradas matriciais negativas

Podemos observar um fenômeno recorrente em nossos resultados: a aparição de valores negativos nas entradas de  $S$ , as quais até mesmo se distribuem com certa simetria em torno da origem, como vemos nos gráficos das Subfiguras 15(b) e 15(e). Isso não se verifica de maneira sistemática nas entradas de  $L$ , como mostram as Subfiguras 15(c) e 15(f). Alguns aspectos são importantes para entender esse comportamento. De partida, temos que a escolha do valor de  $t$  na função objetivo do nosso modelo matemático (Problema (2.7)) castiga mais a norma nuclear de  $L$  – e, indiretamente, seu posto – do que a norma 1 e a esparsidade de  $S$ . Assim, de certa forma, faz sentido pensar que  $L$  é a matriz privilegiada na resolução do problema e isso pode justificar a tendência da matriz de posto se manter em  $[0, 1]^{m \times n}$ , mesmo sem uma restrição explícita no modelo matemático. Portanto, dado que os elementos de  $A$  e de  $L$  pertencem a  $[0, 1]$ , a restrição de factibilidade impede que os elementos de  $S$  escapem do intervalo  $[-1, 1]$ .

Sob o ponto de vista da aplicação, o que determina o sinal negativo de algumas entradas de  $S$  é a relação entre o tom de cinza do plano de fundo e o do objeto móvel que o encobre. Por simplicidade, pensemos a nível de píxel. Suponhamos que  $x$ ,  $y$  e  $z$  sejam entradas atreladas a um mesmo píxel das matrizes  $S$ ,  $L$  e  $A$ , nessa ordem, e que sabidamente  $y$  e  $z$  pertençam a  $[0, 1]$ . Como a exigência de factibilidade impõe  $x + y = z$ , um valor negativo para  $x$  em  $S$  ocorre se, e somente se,  $z - y < 0$ , ou

seja,  $z < y$ . Em termos de tons de cinza, a desigualdade  $z < y$  significa que o píxel registrado no vídeo original ( $A$ ), quantificado por  $z$ , é mais escuro do que o píxel correspondente do plano de fundo, de intensidade  $y$ . Portanto, entradas negativas em  $S$  tendem a ser consequência de objetos móveis mais escuros, em todo ou em parte, do que o plano de fundo na cena em questão.

Matematicamente, a ocorrência de entradas negativas em  $S$  é consequência direta da maneira escolhida para decompor a matriz  $A$  em outras duas no modelo: por meio da soma. Note que o processo que ocorre na captura das imagens quando um objeto se interpõe entre a câmera e o plano de fundo não corresponde a uma soma, mas sim a uma substituição de valores regida justamente por essa interposição, ou seja, pelo quê está à frente ou atrás. Ainda assim, o uso da soma na modelagem se justifica porque as operações de adição e de substituição de dois píxeis, quando um deles é nulo, são equivalentes – e  $S$ , por hipótese, tem o zero na maioria de suas entradas.

### 6.3.2 Valores Singulares

As Figuras 16 e 17 ilustram as distribuições dos valores singulares das matrizes dos vídeos, separadas de acordo com o grupo ao qual pertencem. A Figura 16 exhibe todos os valores singulares de  $A$ ,  $S$  e  $L$ , tanto em escala linear (linha de cima) quanto em escala logarítmica (linha de baixo). Já a Figura 17 mostra apenas os 10 maiores valores singulares de cada matriz, em escala linear. Destacamos que estão representados todos os valores singulares retornados pela função nativa `svd` do MATLAB com o parâmetro `'econ'` da versão reduzida da decomposição. Note também que as instâncias têm dimensões diversas (cf. Tabelas 13 e 14), o que faz variar a quantidade possível de valores singulares de uma instância para outra.

Olhando para os gráficos das colunas à esquerda, nas Figuras 16 e 17, vemos uma tendência geral do maior valor singular da matriz de dados  $A$  se destacar dos demais, em magnitude. A transição entre o primeiro e o segundo valor singular é mais uniforme entre as instâncias do grupo CAVIAR do que entre as do grupo SBI. Essas são características dos próprios vídeos digitais dos conjuntos de teste e podem ser atribuídas justamente à ocorrência de planos de fundo quase estáticos ao longo de cada gravação.

Por outro lado, as distribuições dos valores singulares das soluções ( $S$ ,  $L$ ) de referência estão atreladas à capacidade do modelo matemático em realizar a separação, assim como à aptidão do ADMM em resolvê-lo. Pelas colunas da direita das Figuras 16 e 17, vemos que o maior valor singular de  $L$  em cada instância é muito próximo do maior valor singular de  $A$  correspondente, o que corrobora a afirmação sobre a influência dos planos de fundo nos maiores valores singulares de  $A$ . Contudo, vemos que os valores singulares de  $L$  em cada instância decrescem até certo valor e saltam

para um patamar horizontal que, na prática, representa o valor zero e indica até qual posição considerar na determinação do posto da matriz (veja as colunas sobre o posto nas Tabelas 13 e 14). A média do posto percentual entre as instâncias de CAVIAR é semelhante à de SBI: 47% e 49% da quantidade de colunas, respectivamente. No entanto, os postos percentuais no grupo CAVIAR desviam mais da média do que os do grupo SBI: o desvio padrão no primeiro é de 9 pontos, enquanto no segundo é de apenas 2. Esses dados são os mesmos independente da coluna de posto que utilizamos das tabelas.

Por fim, como vemos nas colunas centrais dessas figuras, os valores singulares de  $S$  tendem a ser de ordem de grandeza menor do que os de  $L$  nas primeiras posições. Nas instâncias do grupo SBI, essa tendência se mantém e os valores singulares de  $S$  decrescem com a posição que ocupam, se estabilizando em determinado patamar não nulo, como vemos na Subfigura 16(e), enquanto aqueles das matrizes  $S$  do grupo CAVIAR saltam a “zero” perto das últimas posições, na Subfigura 16(b). A exceção é a instância CAVIAR1, de cuja matriz  $S$  os últimos valores singulares também são virtualmente nulos; essa instância foi extraída do mesmo projeto que as do outro grupo, embora a instância CAVIAR2 não tenha apresentado o mesmo comportamento. Curiosamente, podemos constatar que, em geral, as instâncias do grupo CAVIAR têm matrizes  $S$  (e  $A$ ) que podem ser consideradas numericamente singulares, o que quase não se verifica com o grupo SBI (cf. Tabelas 28 e 30).

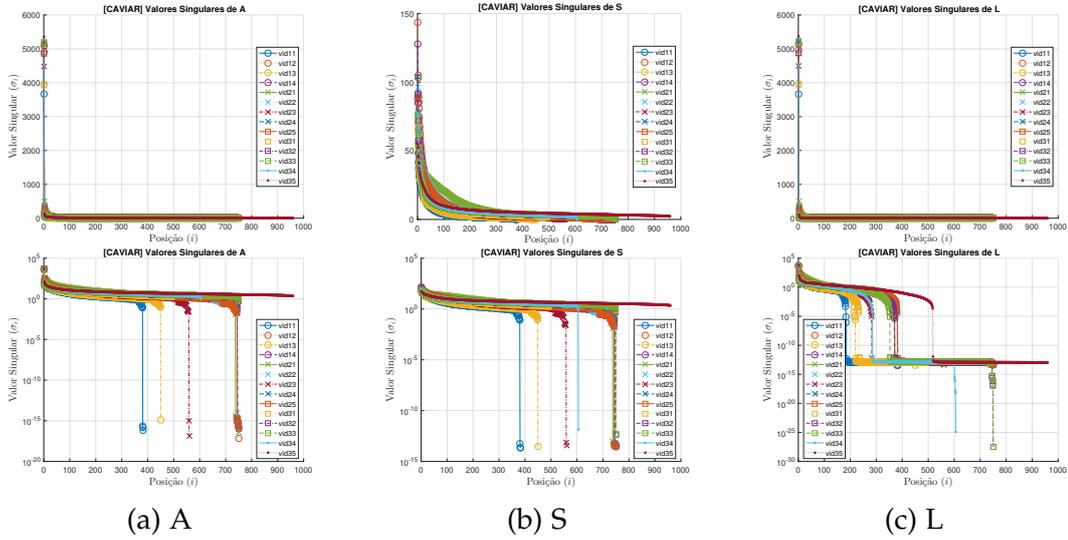
### 6.3.3 Comparação com Matrizes Aleatórias

A partir do que foi discutido nas seções anteriores sobre as entradas matriciais e os valores singulares das instâncias dos vídeos, podemos traçar uma comparação entre as instâncias práticas e as aleatórias, estas discutidas no Capítulo 4. Em especial, retomaremos as notações lá utilizadas e as Figuras 3 e 4.

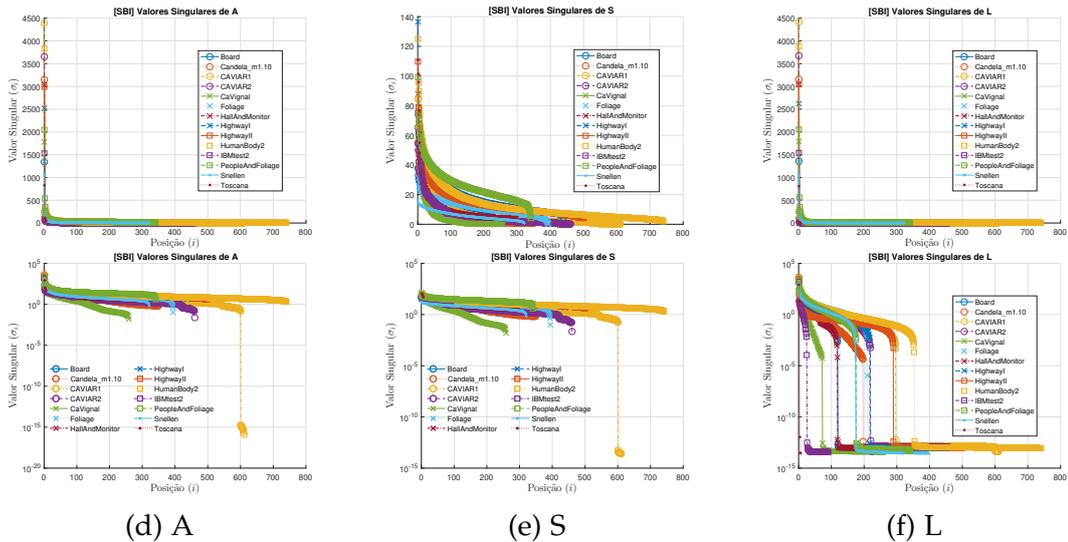
Como esperado, as instâncias dos vídeos apresentaram características próprias e, de maneira geral, distintas das aleatórias. Mesmo impondo tolerâncias numéricas sobre o que consideramos um valor computacionalmente equivalente a zero, as soluções de referência dos vídeos apresentam em  $S$  taxas de esparsidade bem maiores do que o teto de 10% das aleatórias, assim como as matrizes  $L$  têm posto percentual consideravelmente superior aos também 10% que limitaram o posto das instâncias artificiais. Tais informações sobre o posto e a esparsidade das matrizes de entrada e soluções de referência podem ser consultadas nas Tabelas 27 a 30, disponibilizadas no Apêndice D, assim como fizemos com suas equivalentes aleatórias.

Para além da esparsidade de  $S$  e do posto de  $L$ , também as distribuições se mostraram diferentes. No que diz respeito às entradas da matriz  $A$ , os valores nas instâncias dos vídeos, sempre positivos, estão mais uniformemente distribuídos em

Figura 16 – Valores Singulares das Instâncias dos Vídeos  
Instâncias do Grupo CAVIAR



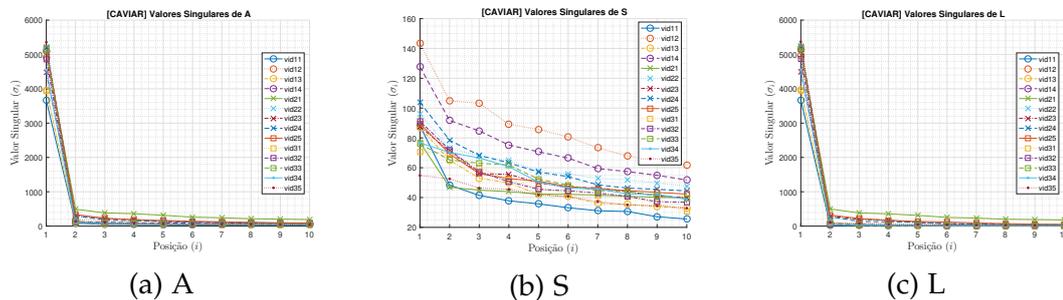
Instâncias do Grupo SBI



seu intervalo unitário do que ocorria com as matrizes aleatórias – basta ver a incidência de *outliers* em cada matriz (cf. Subfiguras 15(a), 15(d), 3(a), 3(d) e 3(g)). Nas matrizes  $A$  aleatórias, de Subfiguras 3(a), 3(d) e 3(g), além da simetria em torno do zero, havia o caso extremo de Candès *et al.* em que era possível distinguir com clareza, apenas pelas magnitudes das entradas de  $A^{CLMW}$ , quais eram exatamente os elementos afetados pelos impulsos de  $S^{CLMW}$  (cf. Subfigura 3(a)). Isto talvez justifique a quantidade ínfima de iterações realizadas pelo ADMM para resolver tanto as instâncias de menor porte quanto as de milhões de variáveis (cf. Tabelas 3 a 5).

Adicionalmente, não surpreende o fato de que a impulsividade característica de  $S^{Yimp}$  e  $S^{CLMW}$  (Subfiguras 3(e) e 3(b)) não se verifica nas matrizes dos vídeos (Subfiguras 15(b) e 15(e)). Neste aspecto, as instâncias gaussianas de Yuan e Yang são mais coerentes com a aplicação que visamos neste trabalho, apesar das diferenças nas

Figura 17 – Maiores Valores Singulares das Instâncias dos Vídeos  
Instâncias do Grupo CAVIAR

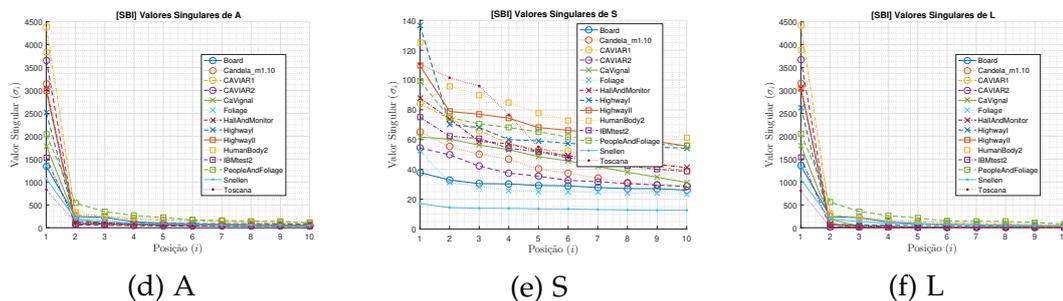


(a) A

(b) S

(c) L

Instâncias do Grupo SBI



(d) A

(e) S

(f) L

amplitudes da distribuição (cf. Subfiguras 3(h), 15(b) e 15(e)).

Quanto aos valores singulares, iniciando por  $L$ , a diferença que mais chama a atenção é o destaque em magnitude que o primeiro valor singular tem com relação aos demais nas instâncias dos vídeos (Subfiguras 17(c) e 17(f)), o que não ocorre nas instâncias aleatórias (Subfiguras 3(c), 3(f) e 3(i)). É como se, na aplicação de que tratamos, um modelo de posto unitário para o plano de fundo se destacasse, sem no entanto desprezar suas modificações secundárias – as demais correções de posto 1 advindas da SVD –, já que os valores singulares subsequentes estão longe de serem desprezíveis. De fato, há uma tendência de que, com exceção do primeiro e daqueles que podem ser considerados numericamente nulos, os valores singulares de  $L$  nessas instâncias sejam da mesma ordem de grandeza dos valores singulares correspondentes de  $S$ , assim como também dos de  $A$  (cf. gráficos em cada linha da Figura 16).

Já nas instâncias aleatórias, as correções de posto um de  $L$  têm normas euclidianas que decrescem gradativamente sem grandes saltos (cf. Subfiguras 4(c), 4(f), 4(i)). Uma interpretação disso sob a perspectiva da aplicação seria a de que, mesmo na situação de  $L$  com posto baixo e quantidade predeterminada de valores singulares, os planos de fundo dos *frames* de cada instância gerada aleatoriamente variam mais do que o aceitável para poderem ser *bem aproximados* apenas pela primeira correção de posto 1 da SVD; as contribuições das demais correções de posto 1 causam impacto comparável ao da primeira, no sentido da norma euclidiana. Vale lembrar que, nas normas euclidiana e nuclear, o erro cometido ao aproximar uma matriz  $L$  de posto  $r$  e

valores singulares  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$  com a melhor aproximação de posto  $k < r$  é igual a  $\sigma_{k+1}$  e  $\sum_{i=k+1}^r \sigma_i$ , respectivamente (veja o Teorema 2).

Dessas observações, podemos constatar que nenhum dos padrões de construção aleatórios reflete bem a distribuição de entradas e valores singulares das matrizes de instâncias práticas. No entanto, o que melhor o representa é o padrão gaussiano de Yuan e Yang, pela semelhança na dispersão das entradas de  $S^{YY_{\text{gaus}}}$  e pelo destaque dos valores singulares  $L^{YY_{\text{gaus}}}$  sobre os  $A^{YY_{\text{gaus}}}$ , que ocorre com o primeiro valor singular nas instâncias práticas. Por outro lado, o padrão mais discrepante em comparação com as instâncias de aplicação é o de Candès *et al.*, já que as posições das entradas de  $A^{\text{CLMW}}$  afetadas pelos impulsos de  $S^{\text{CLMW}}$  são facilmente determinadas devido às discrepâncias de suas magnitudes, e também porque os valores singulares de  $L^{\text{CLMW}}$  são ofuscados pelos de  $S^{\text{CLMW}}$  na formação de  $A^{\text{CLMW}}$  – ou seja, os papéis desempenhados por  $S$  e  $L$ , nesses sentidos, se invertem dessas instâncias para as práticas.

## 7 COMPARAÇÃO DE DESEMPENHO NAS INSTÂNCIAS DOS VÍDEOS

Neste capítulo, apresentamos os principais resultados do nosso estudo comparativo dos desempenhos do SPGM, introduzido neste trabalho, e do ADMM na versão de Candès *et al.* ([21]) e Yuan e Yang ([103]), em instâncias práticas de videovigilância.

Propomos uma configuração simplificada para o SPGM aspirando uma precisão suficiente para corresponder a imagens de boa qualidade no Espaço de Exibição e, nesse sentido, mostramos que o método supera o desempenho do ADMM por resolver mais problemas a um custo consideravelmente menor. Em outras palavras, mostramos que o desempenho do SPGM qualifica a combinação da estratégia proximal com a de continuação homotópica como eficiente na resolução de problemas práticos de decomposição posto-esparsidade (SLRMD), quando não há necessidade de explorar configurações de alta precisão.

Com a introdução das medidas comparativas da Seção 5.4, computadas no Espaço de Exibição, constatamos também que a estimativa de percepção visual por nós estabelecida se revela suficiente para obter imagens de boa qualidade visual.

O itinerário deste capítulo consiste no seguinte. Inicialmente, apresentamos os resultados dos experimentos computacionais com as instâncias dos vídeos, ou seja, comparamos os erros relativos obtidos com o ADMM e o SPGM, fundamentalmente por meio de gráficos. Após discutir esses dados, os condensamos por meio dos *data-profiles* com o objetivo de delinear conclusões a partir dos sucessos e fracassos em atingir a precisão-alvo. Por fim, nos valem das medidas definidas no Espaço de Exibição para confrontar os resultados recém-obtidos.

### 7.1 Erros Relativos às Soluções de Referência

Para comparar o desempenho do SPGM com o do ADMM nas instâncias práticas de videovigilância, executamos ambos os métodos nos problemas de decomposição posto-esparsidade com matrizes oriundas dos dois grupos de filmagens descritos no Capítulo 6, no total de 28 instâncias divididas entre o grupo CAVIAR e SBI, seguindo as configurações dadas pelas Tabelas 15 e 16.

Lembramos que o primeiro grupo é composto por 14 dos 54 vídeos de teste

Tabela 15 – Parâmetros de Configuração do ADMM na resolução dos problemas de videovigilância

$$S^{(0)} = L^{(0)} = Z^{(0)} = 0_{n \times n} \quad \rho = \frac{n^2}{4 \|\text{vec}(A)\|_1} \quad M_{\text{it}} = 1000 \quad \epsilon_{\text{relinf}} = -\text{Inf}$$

Tabela 16 – Parâmetros de Configuração do SPGM na resolução dos problemas de videovigilância

$$\begin{array}{ccccc} T = 4 & \mu \in 10^{\{-4, -3, -2, -1\}} & M_{\text{it}} = 1000 & \sigma = 10^{-4} & L^{(0)} = 0_{n \times n} \\ \epsilon_x = 10^{-6} & \alpha_{\text{min}} = 10^{-30} = 1/\alpha_{\text{max}} & M_{\text{itLS}} = 20 & \zeta = 0.5 & M = 20 \end{array}$$

do Projeto CAVIAR [23], selecionados por nós de maneira a contemplar circunstâncias diversas nas duas situações de gravação consideradas no projeto. O segundo grupo é composto por todos os 14 vídeos do conjunto de testes SBI [88], mantido por Maddalena e Petrosino [67, 12] com o objetivo de padronizar os procedimentos desse tipo de experimento.

As duas principais medidas analisadas nesta etapa são o número de decomposições em valores singulares (nSVD), que mede o custo para os métodos computacionais obterem seus iterandos, e o Erro Relativo (ER) – um desvio percentual do iterando à solução de referência, medido na norma de Frobenius matricial –, o qual quantifica a distância ao ponto alvo, no sentido da Definição 4.1.

Vale retomar também que, conforme estabelecemos na Seção 5.3, consideramos como solução de referência um ponto viável obtido a partir de uma execução de longo prazo ou baixa infactibilidade do ADMM sucedida pela viabilização de menor valor objetivo para a função objetivo original (2.7).

As Figuras 18 e 19 contêm os gráficos de erro relativo por número de decomposições do tipo SVD gastas pelo ADMM e pelo SPGM com os vídeos do primeiro e do segundo grupo, respectivamente, com eixo vertical em escala logarítmica.

Observe inicialmente que há uma diferença entre os comprimentos das curvas de cada um dos métodos. Isso ocorre porque forçamos o ADMM a realizar mil iterações a fim de evitar critérios que o prejudicassem na comparação e, ainda, para não esconder seu comportamento de mais longo prazo. Por outro lado, o horizonte de execução do SPGM está relacionado com a precisão final almejada pelo método, a qual é estabelecida *a priori*. De fato, tal precisão final é idealizada por meio da escolha da sequência de parâmetros de suavização (quantos e quais) e também do nível de precisão com que cada subproblema homotópico é resolvido (tolerância no tamanho do passo).

Figura 18 – Gráficos de erros relativos ao longo das iterações do ADMM e do SPGM para os vídeos do Grupo CAVIAR, seguindo as configurações das Tabelas 15 e 16. Os pequenos quadrados nas curvas do SPGM indicam o fim de uma iteração homotópica, ou seja, o término da resolução de um subproblema, durante a qual o parâmetro de suavização é mantido fixado.

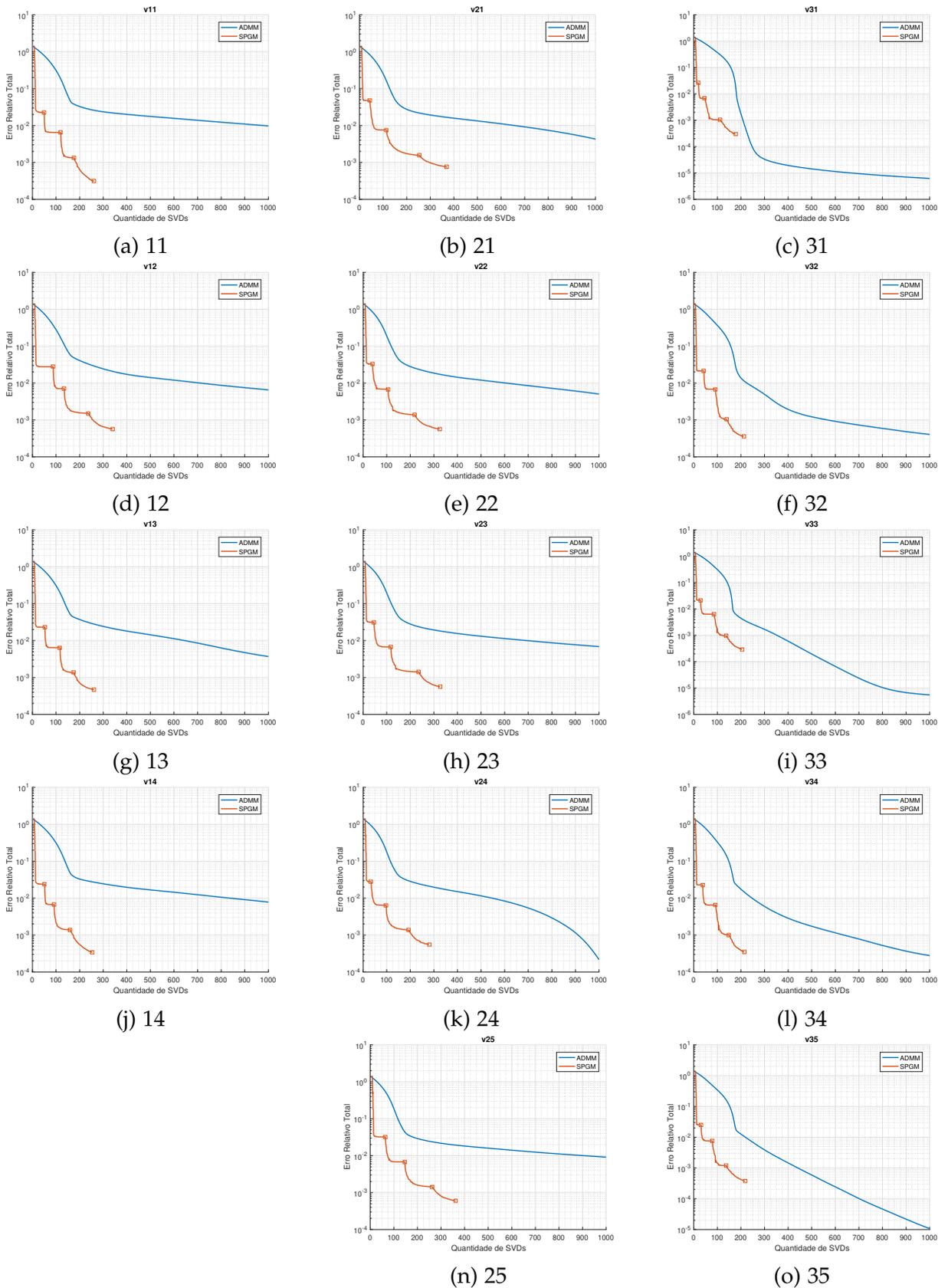
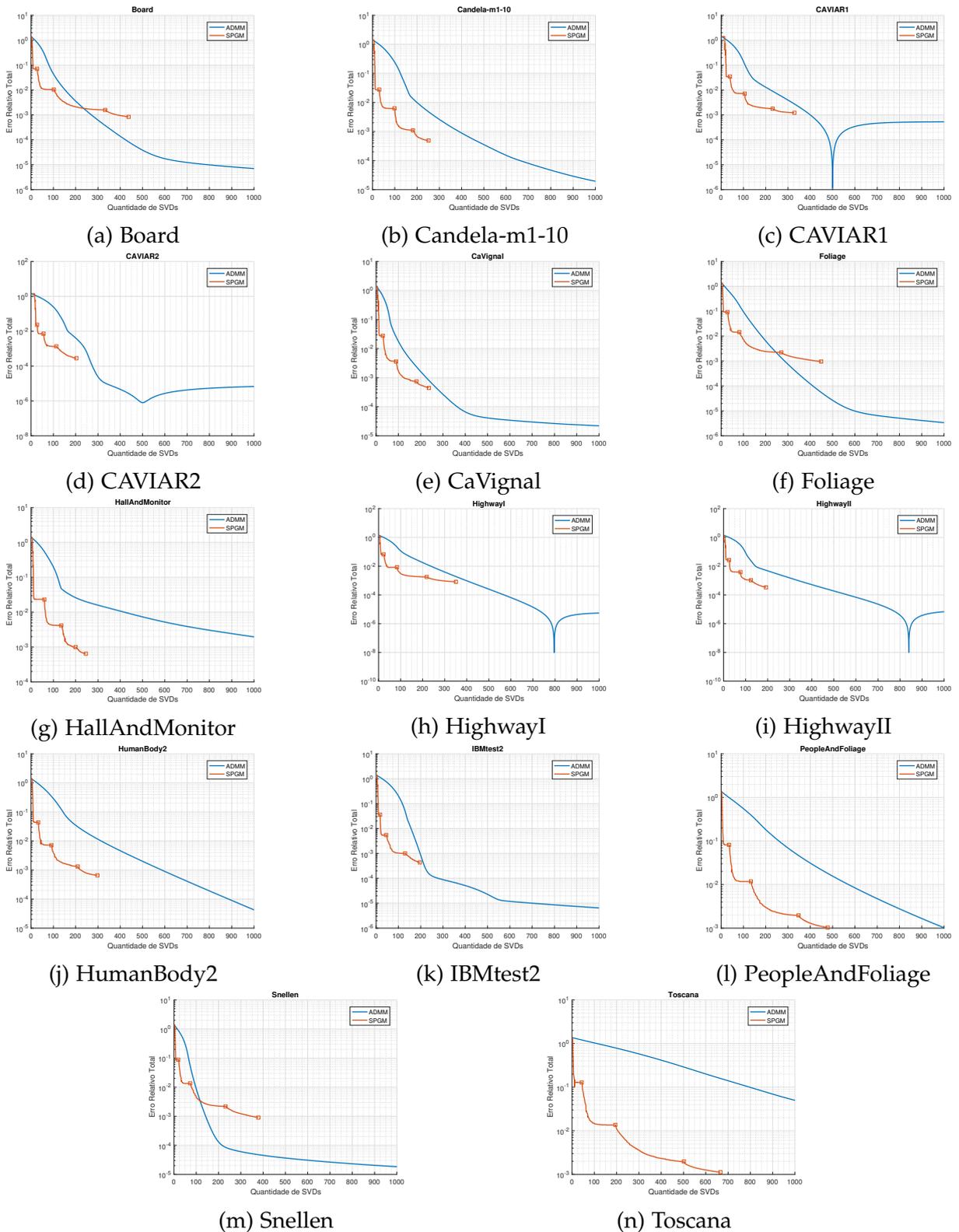


Figura 19 – Gráficos de erros relativos ao longo das iterações do ADMM e do SPGM para os vídeos do Grupo SBI, seguindo as configurações das Tabelas 15 e 16. Os pequenos quadrados nas curvas do SPGM indicam o fim de uma iteração homotópica e a consequente mudança no valor do parâmetro de suavização caso ainda restem etapas. As inversões na tendência de decrescimento das curvas do ADMM indicam a ultrapassagem da solução de referência pelos iterandos.



No nosso caso, em que miramos na precisão final de um milésimo para o erro relativo, é normal que a execução não se prolongue muito pois o tamanho do passo tende a se tornar suficientemente pequeno próximo à solução do último subproblema. Outra situação que encurta o processo é quando os iterandos passam a se mover muito lentamente, ainda que não estejam nas proximidades da solução, e ocorre a parada por estagnação. Há ainda que se considerar o fato de que os subproblemas homotópicos, mesmo quando bem resolvidos, podem não refletir a precisão almejada simplesmente por se tratarem de aproximações para o problema original.

### Curvas de Erro: SPGM

Além do comprimento das curvas, há também distinções em seus formatos. Como é característico do SPGM, o decréscimo no erro é mais brusco nas primeiras iterações de cada subproblema homotópico; adicionalmente, com o passar das iterações homotópicas, a queda no erro é amenizada pela crescente dificuldade numérica em conquistar precisão. Vale ressaltar que a etapa de busca linear no SPGM pode demandar mais de uma SVD por iteração e isso se reflete na curva de erro como um segmento de reta horizontal, já que somente após terminada a busca é que se conquista um novo valor para o erro relativo (veja, por exemplo, o pequeno trecho inicial da curva do SPGM na Subfigura 19(d)). Entretanto, na prática, os resultados mostram que dificilmente se gastou mais de uma SVD por iteração, o que pode ser constatado pela comparação entre os valores  $n_{\text{Iter}}$  e  $n_{\text{SVD}}$  nas Tabelas 31 e 32. Sendo assim, o aspecto geral das curvas de erro do SPGM com  $T = 4$  iterações homotópicas é o da ligação sequencial de quatro trechos de queda inicial acentuada seguida por estabilização tendente a um patamar horizontal.

Percebemos pelos gráficos que a queda do erro nas primeiras iterações é vertiginosa, embora logo se consolide num patamar quase horizontal. Essa estabilização do erro é um indício de que o subproblema já está praticamente resolvido na precisão que lhe cabe e, como consequência, os passos se tornam cada vez menores a partir daí. No momento em que o tamanho do passo fica abaixo da tolerância do subproblema, a iteração homotópica é declarada encerrada e uma outra se inicia com um novo valor para o parâmetro de suavização (um décimo do anterior, no caso destes experimentos), tendo como ponto inicial o último iterando do subproblema imediatamente anterior. Por sua vez, o novo subproblema aproxima melhor o problema original e, conseqüentemente, sua solução conduz a uma nova queda no erro.

A cada recomeço da etapa homotópica, portanto, a queda no erro volta a se acentuar por algumas iterações, nem sempre com a mesma rapidez, mas inevitavelmente tendendo a um novo patamar horizontal logo em seguida. Esse processo se repete de maneira semelhante a cada iteração homotópica. Com a diminuição sucessiva

do parâmetro de suavização, o SPGM vai se aproximando da solução do problema original e o que determina a precisão desejada é o último (e menor) valor desse parâmetro. Pelos gráficos, fica evidente que é nas primeiras iterações homotópicas que o erro decresce mais e de maneira mais rápida.

Assim como nos casos aleatórios, o valor de  $\mu$  nestes experimentos foi reduzido décimo a décimo de  $10^{-1}$  até  $10^{-4}$  e, portanto, existia *a priori* uma limitação imposta ao erro mínimo que o SPGM poderia obter, a qual é dada pela qualidade da aproximação da função original no último subproblema.

De uma forma geral, podemos observar que o último ciclo homotópico se prolonga por mais iterações sem trazer os benefícios de precisão dos ciclos anteriores. Isso nos dá uma ideia do que ocorreria caso continuássemos a realizar mais ciclos homotópicos nesse mesmo esquema: muitas SVDs a mais e pouca melhora na qualidade da solução. De fato, simulações nessa direção, não detalhadas aqui, conduziram ou a longos patamares horizontais ou a curtíssimos ciclos homotópicos sem diminuição do erro.

Uma pergunta que permanece em aberto é se é possível calibrar a sequência de parâmetros de suavização (quantos e quais) e os critérios de troca de iteração homotópica de maneira que se consiga obter mais quedas vertiginosas e patamares horizontais cujo prolongamento não comprometa tanto o orçamento em número de SVDs.

### Curvas de Erro: ADMM

Por sua vez, as curvas de erro do ADMM tipicamente se iniciam decaindo em formato côncavo para, em seguida, adquirirem uma forma convexa, aproximadamente afim, que tende a predominar durante a maior parte do tempo de execução. Isso mostra que a velocidade de convergência é maior no início das iterações, com aspectos de convergência superlinear (quadrática), e que, passada a inflexão, o decaimento do erro se torna mais lento. Como vemos em quase todos os cenários, esse ponto que marca o início da lentidão ocorreu cedo – com menos de duzentas iterações, antes do término do SPGM – de forma que o decréscimo inicialmente acelerado não foi suficiente para conquistar os erros relativos do método espectral com orçamento semelhante. Dois exemplos representativos são os vídeos 14 (Subfigura 18(j)) e HallAndMonitor (Subfigura 19(g)).

Destoam parcialmente das demais as instâncias CAVIAR1 (Subfig. 19(c)), CAVIAR2 (Subfig. 19(d)), HighwayI (Subfig. 19(h)) e HighwayII (Subfig. 19(i)), nas quais um pouco após a inversão de concavidade podemos observar que o erro volta a crescer. Isso se deve ao fato de que a sequência de iterandos em certo momento passou

pela solução de referência sem que a execução fosse terminada, já que o critério de parada adotado agora foi o do número exato de mil SVD, e não aqueles utilizados na obtenção das soluções de referência, os quais nesses casos foram satisfeitos antes de esgotar esse orçamento. Vale destacar que o crescimento ao final das iterações se mostrou limitado a um patamar assintótico relativamente baixo, o que sugere que a sequência não se afastou consideravelmente da solução de referência. Nos demais casos, para se ter ideia de quanto tempo a lenta convergência se mantém basta ver os gastos na obtenção das soluções de referência nas Tabelas 13 e 14.

Considerando os aspectos discutidos aqui, nossos resultados estão de acordo com as expectativas para o desempenho de um método de primeira ordem, como é o caso do ADMM [16, 41, 77].

## 7.2 Convergência Segundo a Precisão-alvo

### Os Menores Erros

O aspecto mais relevante a ser destacado nos gráficos das Figuras 18 e 19 é a diferença entre as taxas de decrescimento dos erros do SPGM e do ADMM. Em praticamente todas as instâncias fica evidente que a queda com o SPGM é consideravelmente mais íngreme do que com o ADMM nas primeiras centenas de SVD gastas. Além disso, com exceção de três casos – Board (Subfig. 19(a)), Foliage (Subfig. 19(f)) e Snellen (Subfig. 19(m)) –, a curva do SPGM se localiza completamente abaixo da curva do ADMM. Isso indica que, para a faixa de orçamento que envolve o gasto do SPGM, o método espectral obteve resultados consistentemente melhores e que manteria esse desempenho superior ainda que o orçamento do usuário viesse a sofrer reduções.

Mesmo para os vídeos do grupo SBI, com os quais se verifica uma competitividade um pouco mais acirrada, o menor erro para cada número de SVDs gasto é quase sempre do SPGM. As exceções são os gastos com a primeira iteração, como explicado há pouco, e as instâncias Board (Subfig. 19(a)), Foliage (Subfig. 19(f)) e Snellen (Subfig. 19(m)), em que as curvas de erro se cruzam em certa iteração a partir da qual a ordem se inverte: o ADMM passa a ter erro menor do que o SPGM. No entanto, essas três instâncias representam apenas 11% dos casos totais considerados, o que sustenta a constatação de melhor desempenho do SPGM na situação em que o orçamento em número de SVDs é restrito.

### Precisão-alvo

Tendo em vista o objetivo de conseguir erro relativo menor do que ou igual a  $1.0 \times 10^{-3}$ , estabelecido na Seção 5.5, organizamos a Tabela 17 com os dados de quais

problemas foram resolvidos com tal precisão por cada um dos dois *solvers*.

Tabela 17 – Indicação de quais problemas cada *solver* foi capaz de resolver com erro relativo menor do que ou igual a  $1.0 \times 10^{-3}$ , seguindo os critérios de parada apresentados. O símbolo  $\checkmark$  indica que a precisão-alvo foi atingida, enquanto  $\times$  expressa o contrário.

Instância	SPGM	ADMM	Intância	SPGM	ADMM
11	$\checkmark$	$\times$	Board	$\checkmark$	$\checkmark$
21	$\checkmark$	$\times$	Candela_m1.10	$\checkmark$	$\checkmark$
31	$\checkmark$	$\checkmark$	CAVIAR1	$\times$	$\checkmark$
12	$\checkmark$	$\times$	CAVIAR2	$\checkmark$	$\checkmark$
22	$\checkmark$	$\times$	CaVignal	$\checkmark$	$\checkmark$
32	$\checkmark$	$\checkmark$	Foliage	$\checkmark$	$\checkmark$
13	$\checkmark$	$\times$	HallAndMonitor	$\checkmark$	$\times$
23	$\checkmark$	$\times$	HighwayI	$\checkmark$	$\checkmark$
33	$\checkmark$	$\checkmark$	HighwayII	$\checkmark$	$\checkmark$
14	$\checkmark$	$\times$	HumanBody2	$\checkmark$	$\checkmark$
24	$\checkmark$	$\checkmark$	IBMtest2	$\checkmark$	$\checkmark$
34	$\checkmark$	$\checkmark$	PeopleAndFoliage	$\times$	$\times$
25	$\checkmark$	$\times$	Snellen	$\checkmark$	$\checkmark$
35	$\checkmark$	$\checkmark$	Toscana	$\times$	$\times$
SUCESSOS:	14	6	SUCESSOS:	11	11

Constatamos tanto pelas Figuras 18 e 19 quanto pela Tabela 17 que o SPGM obteve sucesso em 25 das 28 instâncias (89%), enquanto que os êxitos do ADMM nesse mesmo total ocorreram em apenas 17 instâncias (61%). Note que, dos dois únicos vídeos em que o SPGM não alcançou a precisão almejada, CAVIAR1 (19(c)) e Toscana (19(n)), um deles também está na lista de insucessos do ADMM.

Vale ressaltar que as duas instâncias que nenhum dos dois *solvers* foi capaz de resolver foram CAVIAR1 e Toscana. Conforme vimos no Capítulo 6, os dois vídeos do projeto CAVIAR presentes no conjunto SBI apresentaram dificuldades desde a obtenção de uma solução de referência. Na ocasião, foi preciso impor um número mínimo de iterações ao ADMM para que ele não parasse já nas primeiras iterações com infactibilidade relativa da ordem de  $10^{-11}$  porém com baixíssima qualidade visual de separação do plano de fundo. Mesmo assim, ao contrário do que ocorreu com CAVIAR2, a quebra da matriz da solução de referência de CAVIAR1 ainda foi de pouca qualidade visual. Trata-se de uma cena com dinâmica mais complexa em que vários elementos estão se movendo - alguns entram em cena e ficam estagnados, outros param por um certo período e voltam a se mover -, o que impõe mais dificuldades à identificação do que realmente é plano fundo. Por sua vez, a cena TOSCANA é composta por apenas seis *frames* tomados em horários distintos e que mostram pedestres caminhando, características que poderiam justificar os insucessos.

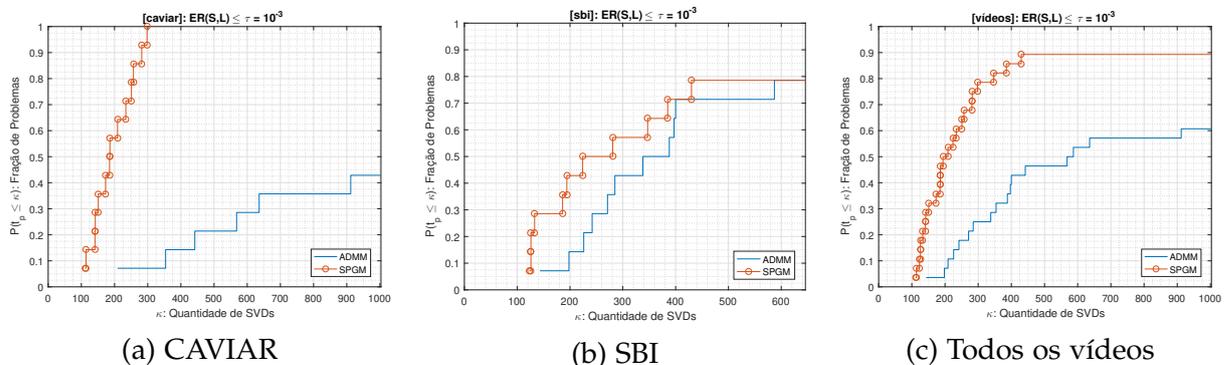
### Condensando Resultados

Uma vez estabelecido o critério de atingir erro relativo igual ou inferior a  $10^{-3}$ , podemos sintetizar visualmente os resultados apresentados nas Figuras 18 e

19 e na Tabela 17 por intermédio de um gráfico que registra a fração de problemas resolvidos – no sentido do critério instituído – em função da quantidade de recursos gastos, em número de decomposições SVD. A Figura 20 é baseada nos *data profiles* introduzidos por Moré e Wild em seu trabalho sobre comparação de métodos de otimização sem derivada ([70]), e separa os dados de acordo com os três grupos de problemas. Em seu contexto original, assim como aqui, o usuário dispõe de recursos escassos em relação ao tamanho dos problemas e, portanto, é conveniente que a análise de desempenho se baseie em ferramentas gráficas que apresentem resultados em função do custo de resolução do problema.

Fica evidente, pela Figura 20, que seja qual for o orçamento do usuário, a escolha pelo SPGM é preferível, na medida em que esse método sistematicamente resolve uma fração maior de problemas do que o ADMM.

Figura 20 – Curvas que ilustram o percentual de problemas que cada *solver* consegue resolver, de acordo com o critério que considera a instância resolvida quando o erro relativo atingido é menor ou igual a  $10^{-3}$ . Neste trabalho, essa precisão foi adotada como alvo com o objetivo de gerar imagens em que a separação do plano de fundo é realizada com qualidade visual tão favorável quanto permite a abordagem adotada para o problema convexo original.



### 7.3 Medidas no Espaço de Exibição: ER, PEP, EMPE $\times$ nSVD

Até o momento, temos nos debruçado sobre o erro relativo a fim de quantificar a qualidade dos iterandos e determinar se, em certo sentido, o problema foi suficientemente bem resolvido. Embora a escolha do valor 0.1% como limiar para o erro relativo tenha sido justificada com base no mapeamento que ocorre do espaço de otimização para o de exibição, a medida de erro em si continua se referindo ao espaço no qual o método de otimização atua. Nesta seção, aplicamos as novas maneiras de medir a qualidade dos iterandos no Espaço de Exibição, definidas na Seção 5.4, e apresentamos os resultados associados aos experimentos deste capítulo. Dessa forma,

podemos ver que as novas medidas trazem uma outra perspectiva sobre o que está acontecendo no espaço discreto em que as imagens são exibidas e, igualmente, podemos avaliar o valor limiar para o erro relativo no espaço de otimização que tem sido associado a bons resultados visuais.

#### Motivação de Adoção das Novas Medidas

Quando o problema a ser resolvido tem a finalidade de produzir uma imagem digital a ser exibida em dispositivos de visualização, como em nosso caso, é importante ter em mente que estamos querendo obter uma solução num conjunto discreto, esse a que temos chamado de Espaço de Exibição. Como sabemos, o mapeamento de uma matriz de entradas reais em outra cujos elementos estão restritos a apenas um conjunto limitado de valores inteiros faz com que tenhamos uma margem de erro no espaço contínuo que não altera a imagem exibida – em termos matemáticos, essa transformação, em geral, não é injetora. Portanto, faz sentido pensar que a dinâmica que se verifica em um espaço possivelmente difere daquela que ocorre no outro; é justamente a essa dinâmica que acontece no Espaço de Exibição que nos voltamos agora.

No Espaço de Exibição, pensando em como medir as distâncias entre imagens advindas de iterandos e a imagem da solução, está claro que poderíamos usar as mesmas métricas do espaço de otimização; na prática, porém, isso seria como perder um pouco da sensibilidade às variações no espaço de otimização, não trazendo portanto grandes vantagens.

A chave para pensar no Espaço de Exibição está nas perguntas que podem ser feitas a imagens mas que não faziam tanto sentido no espaço contínuo. Um bom caminho é pensar nos píxeis das imagens correspondentes aos iterandos e compará-los aos da imagem-solução.

- Quantos dos píxeis já atingiram o valor correto?
- Pensando em quantidade de tons da escala de cores, a quantos tons de distância estamos da solução se somarmos todas as discrepâncias entre píxeis correspondentes?
- É possível estimar a maneira como esses tons discrepantes estão dispersos na imagem?

Foi para ajudar a responder esses questionamentos que definimos as medidas no Espaço de Exibição introduzidas na Seção 5.4. A primeira delas, o PEP (Píxeis Errados em Porcentagem), mede qual percentual dos píxeis ainda não atingiu valor igual ao da solução. A segunda é o EMPE (Erro Médio por Píxel Errado), que

representa quantos tons de cinza estão errados, em média, por cada píxel distinto ao da solução. Para recordá-las, consulte a Definição 15.

É importante lembrar que, no cálculo da média, o EMPE ignora completamente os píxeis que já são idênticos aos da solução. Além disso, das definições vistas até aqui, o que inclui o Erro Relativo (ER), decorre que  $ER \in [0, \infty]$ , enquanto  $PEP \in [0, 100]$  e  $EMPE \in [1, 255]$ , o que favorece por exemplo a representação gráfica das duas últimas grandezas em um mesmo eixo, como faremos adiante na exibição dos resultados. Por fim, note que enquanto o PEP está relacionado com a primeira pergunta sobre o Espaço de Exibição, o EMPE dialoga diretamente com as outras duas.

Munidos das novas maneiras de medir o erro no Espaço de Exibição, apresentamos a seguir os resultados dessas medidas nos experimentos de videovigilância.

## Resultados das Novas Medidas

As Figuras 21 e 22 exibem os resultados das medidas ER-L, EMPE-L e PEP-L para cada instância dos grupos CAVIAR e SBI. Optamos por focar nos dados da componente  $L$  principalmente por concisão, sendo a escolha justificada pela semelhança das ordens de grandeza dos erros ER-SL e ER-L – ou seja, porque o erro relativo da componente do posto é o que exerce maior influência no erro relativo total, aquele em que consideramos como variável a justaposição de componentes  $S$  e  $L$ .

Em cada gráfico há dois eixos verticais em escala logarítmica. O eixo da esquerda se refere ao erro no espaço de otimização (ER-L), ao passo que o eixo da direita reúne os “erros” no Espaço de Exibição (PEP-L e EMPE-L). O eixo horizontal, como nos casos anteriores, representa o número de SVDs realizadas.

Para garantir que a leitura dos gráficos está sendo feita de maneira correta, começemos por obter os valores de ER, PEP e EMPE para o ponto inicial, a matriz nula, de onde partem tanto o SPGM quanto o ADMM. As constatações são praticamente as mesmas, independentemente da instância. No eixo à esquerda, quando ainda não se gastou nenhuma SVD, tipicamente temos que  $ER \approx 1$ . Para esse mesmo instante, olhando agora no eixo à direita, é possível ver que, em geral,  $PEP \approx 100$  e  $EMPE \approx 200$ . Isso significa que, no início de tudo, a distância na norma de Frobenius do ponto inicial à solução é igual à norma de Frobenius da própria solução, isto é, há  $ER \approx 1 = 100\%$  de erro no espaço de otimização. Por sua vez, o valor  $PEP \approx 100$  indica que praticamente 100% dos píxeis estão inicialmente errados (esse é seu valor máximo). Já  $EMPE \approx 200$  significa que, em média, cada píxel está errado em algo em torno de 200 tons de cinza – lembre-se: o maior valor possível é de 255.

Apesar da grande quantidade de informação que pode ser extraída dessas figuras e dos padrões que se repetem, alguns aspectos se destacam quando o propósito

Figura 21 – Grupo CAVIAR. Erro Relativo (ER), Erro em Percentual de Píxeis Errados (PEP) e Erro Médio por Píxel Errado (EMPE) para cada uma das instâncias dos vídeos do primeiro grupo.

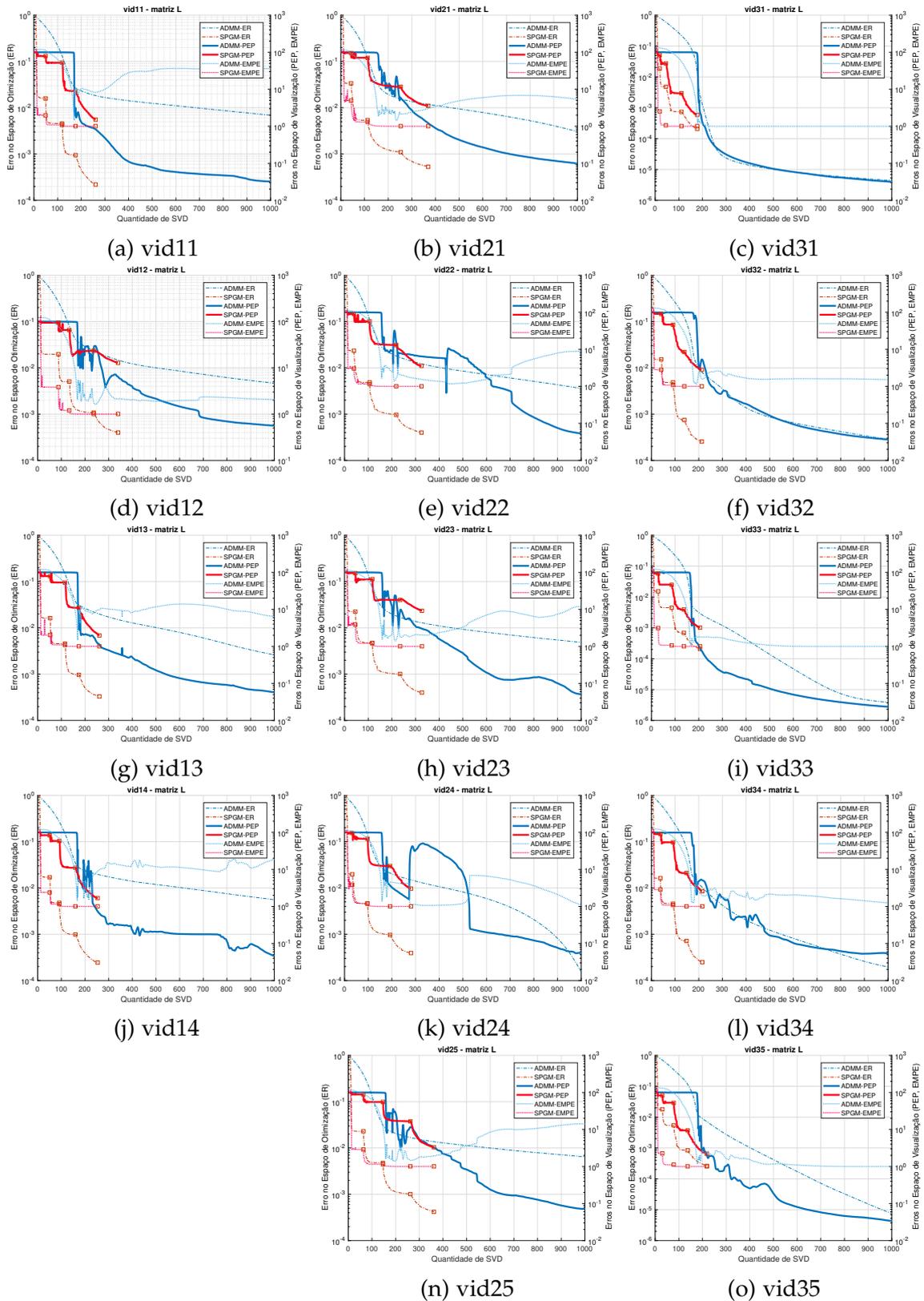
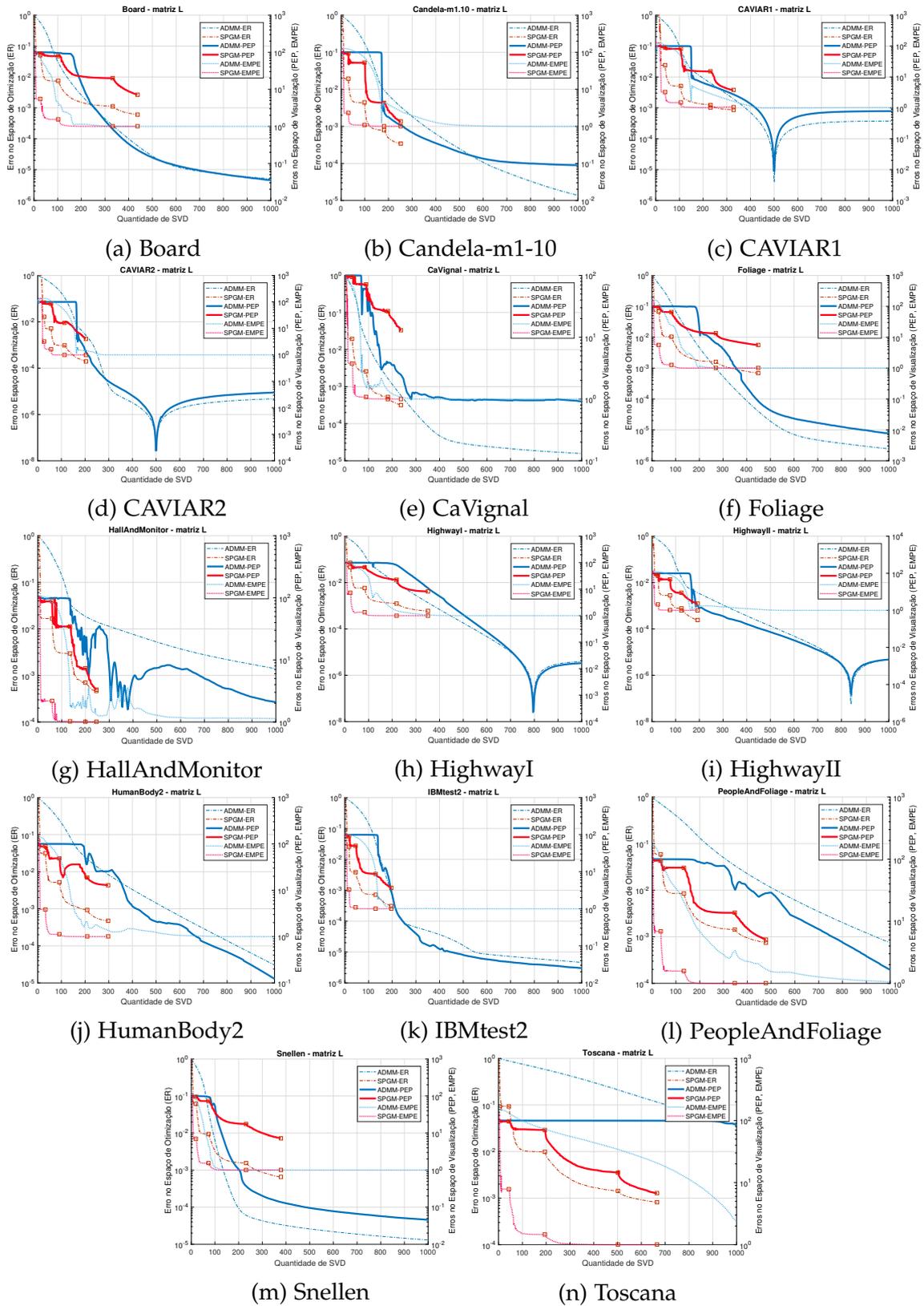


Figura 22 – Grupo SBI. Erro Relativo (ER), Erro em Percentual de Píxeis Errados (PEP) e Erro Médio por Píxel Errado (EMPE) para cada uma das instâncias dos vídeos do segundo grupo.



é a análise comparativa de desempenho dos métodos.

A respeito do SPGM, observamos que a redução no número de píxeis errados (PEP) segue um comportamento semelhante ao do erro relativo (ER) e que, de fato, ao final das iterações, o SPGM termina a execução com poucos pontos percentuais de píxeis errados. Tão ou mais importante que isso é o fato de que o erro médio (EMPE) despenca bruscamente já nas primeiras iterações. Juntando tais constatações e, tendo em mente que o EMPE é uma média do erro em tons de cinza envolvendo apenas os píxeis que ainda estão errados, o que podemos concluir é que o SPGM foi corrigindo todos os píxeis de  $L$  mais ou menos ao mesmo tempo. Além disso, as sucessivas quedas repentinas na curva de PEP espalhadas ao longo das iterações indicam que a cada subproblema homotópico, um subconjunto de píxeis é cravado como correto, a grande maioria nas primeiras iterações do subproblema, e alguns nas demais. No fim, ainda que haja alguns pontos percentuais de píxeis errados, essas entradas matriciais diferem da solução em aproximadamente um tom de cinza, em média, o que possivelmente não compromete a identificação visual da dinâmica da cena.

O ADMM, por sua vez, experimenta um fenômeno diferente. Observe como, em geral, a curva dos píxeis errados (PEP) se mantém constante no início durante uma ou até duas centenas de iterações, para só depois começar um decaimento mais abrupto. O erro médio por píxel (EMPE) tampouco despenca de uma vez, mas sim aos poucos, formando uma curva côncava, semelhante ao que já constatamos anteriormente com seu erro relativo (ER). Embora em várias instâncias o ADMM conquiste a precisão necessária, isso só ocorre após um gasto considerável de SVDs. Na situação hipotética de restrição de orçamento, digamos, a 200 decomposições SVD, a depender da instância, o resultado poderia frustrar os objetivos do usuário. Assim, corroboramos os resultados discutidos exclusivamente a partir dos erros no espaço de otimização.

## 7.4 Considerações Finais

A discussão feita neste capítulo nos conduz à conclusão mais importante sobre o desempenho dos dois métodos nas instâncias práticas de videovigilância: na grande maioria dos casos, o SPGM obtém erro relativo propício à exibição e arquivamento do vídeo antes do ADMM, no sentido da precisão que estabelecemos *a priori* como almejada e em termos do número de SVDs realizadas. Os resultados evidenciam que, sob restrições orçamentárias de capacidade de processamento – e, conseqüentemente, de tempo –, o SPGM é preferível ao ADMM para obter não apenas melhores resultados como também para alcançar semelhança visual com as soluções de referência, nos casos em que esta seja uma tarefa possível de ser realizada pela

abordagem do problema original e com o valor de  $t$  pré-estabelecido.

O contraponto está na desvantagem do SPGM em necessitar de uma configuração de parâmetros para a sequência de subproblemas, algo inexistente no ADMM. Ainda assim, nossos resultados mostram que uma configuração relativamente simples foi eficiente e suficiente para nosso contexto de aplicação, abrindo caminho para a investigação de outras possibilidades de escolha de parâmetros que possam aprimorar o desempenho do método, especialmente com pontos iniciais mais promissores. Dessa forma, concluímos que a combinação da estratégia proximal com a de continuação homotópica, sintetizadas do SPGM, se mostrou robusta e eficiente na resolução do problema SLRMD no contexto de videovigilância.

Por fim, construímos um processo heurístico capaz de melhorar os desempenhos gerais de ambos os métodos e que, com surpreendente eficácia, acelerou notavelmente a convergência do ADMM. Apesar de sua simplicidade, até onde é de conhecimento destes autores, não foi objeto de estudo nos trabalhos por nós investigados.

## 8 DISCUSSÕES SOBRE O PONTO INICIAL

Neste capítulo, discutimos algumas possibilidades para a escolha do ponto inicial e seus impactos no desempenho dos métodos estudados quando da resolução dos problemas aleatórios e de videovigilância, tanto do ponto de vista do problema de otimização quanto da perspectiva da aplicação.

Assim, propomos uma heurística para a determinação do ponto inicial que é atrelada ao problema e que, portanto, pode ser adotada por qualquer método que se dedique a resolvê-lo. Até o momento da preparação deste trabalho, não temos conhecimento de outras publicações que explorem a escolha do ponto inicial no problema de decomposição posto-esparsidade e tampouco suas implicações práticas.

Por fim, investigamos os aspectos visuais dos pontos iniciais no Espaço de Exibição e realizamos comparações em cada problema com a respectiva solução de referência e também com a melhor aproximação de posto unitário, segundo a norma nuclear, para a matriz de entrada.

### 8.1 *Warm-start*: Uma Heurística de Partida Aprimorada

#### Introdução

Até o momento, vimos que o SPGM se mostrou bastante competitivo com relação ao ADMM implementado a partir das diretrizes de Candès *et al.* [21] e Yuan e Yang [103], inclusive superando-o em vários cenários, no sentido dos critérios práticos de exibição que estabelecemos. Nossos resultados mostram a diferença das velocidades de convergência entre os dois métodos, a qual foi evidenciada pela concavidade das curvaturas nos gráficos de erro relativo, principalmente nas primeiras iterações: as curvas do SPGM são consistentemente convexas, enquanto as do ADMM são côncavas.

Como é de se esperar, a fim de reproduzir resultados coerentes com as publicações mencionadas e também para tornar mais direta a sua comparação, temos herdado da versão do ADMM em questão a escolha de matrizes nulas como ponto inicial para as variáveis de otimização<sup>1</sup> em ambos os métodos. No entanto, a própria natureza homotópica do SPGM – no qual o desempenho do método em cada iteração

<sup>1</sup> Convém retomar alguns detalhes. Para o SPGM, que opera sob factibilidade, as variáveis de otimização são as entradas de apenas uma das componentes – neste trabalho, de  $L$  –, enquanto que para o ADMM tanto as entradas de  $S$  como as de  $L$  são variáveis de otimização, sendo a factibilidade uma medida de otimalidade, isto é, a infactibilidade nula ocorre no ponto ótimo. Além disso, vale lembrar que na nossa versão do ADMM a componente esparsa é a primeira a ser atualizada em cada iteração, de forma que, na prática, seu valor inicial é ignorado.

homotópica depende do ponto de partida deixado pela etapa anterior – traz à tona o questionamento a respeito de um ponto inicial mais eficaz do que matrizes nulas.

Nesta seção, apresentamos um processo heurístico de determinação do ponto inicial para o Problema (2.1) que serve para qualquer método cujas variáveis de otimização englobem uma ou ambas as componentes matriciais. Apesar de ser relativamente simples, veremos que essa heurística acelera de maneira surpreendente a convergência do ADMM com relação à versão do método discutida em Candès *et al.* [21] e Yuan e Yang [103], além de garantir ao SPGM a resolução de uma instância a mais dentre as três únicas que não havia resolvido sem o aperfeiçoamento do ponto de partida. Como afirmamos, durante esta pesquisa não foram encontradas publicações que se dedicassem à escolha do ponto inicial, de forma que, até onde é de nosso conhecimento, trata-se de um processo inédito entre trabalhos de decomposição posto-esparsidade.

A ideia por trás da heurística e apresentação do Algoritmo

Vimos na Seção 4.5 que o uso da estratégia de melhor aproximação de posto fixo na métrica da norma nuclear não foi suficiente para resolver o problema SLRMD com erro relativo de ordem menor do que  $10^{-2}$ , mesmo adotando o melhor valor possível para o posto. Na ocasião, exploramos o uso dessa estratégia tanto na componente de esparsidade quanto na de posto, o que também incluiu o caso em que atribuímos a uma das componentes a solução trivial nula, sempre impondo a factibilidade.

O que investigamos agora é se as melhores aproximações de posto fixo não triviais, apesar de não serem por si só soluções precisas, podem funcionar como melhor ponto de partida na resolução iterativa do nosso problema do que os pontos iniciais triviais. Claramente, por serem muitas possibilidades, não testaremos cada uma das aproximações possíveis, como antes, visto que tornaria necessário resolver cada problema centenas de vezes. Com isso, nos deparamos com a dificuldade de escolher apenas uma delas, sendo que, em situações concretas, não dispomos de meios para calcular erros relativos já que a solução é desconhecida.

A ideia da estratégia que propomos é usar a função objetivo do problema original como indicador da qualidade do ponto tentativo, ao qual seguiremos impondo factibilidade. Portanto, diferentemente de antes, não é razoável tentarmos aproximar diretamente a componente  $S$  de esparsidade, já que o simples cálculo da função objetivo acarretaria na realização de mais SVDs. Por outro lado, aproximar a componente matricial  $L$  do posto usando a SVD da matriz  $A$  de entrada não acarretará em custos adicionais de avaliação de função, visto que o cálculo da norma nuclear do ponto tentativo é feito a partir dos valores singulares da aproximação. E é a partir disso

que conceberemos a nossa heurística de determinação do ponto inicial, a qual, na prática, devolve uma aproximação  $L^{(0)}$  para  $L^*$  que pode ser desde a matriz nula até a própria matriz  $A$ , a depender do resultado do processo. O Algoritmo 4 detalha o processo heurístico que desenvolvemos. Note que essa heurística não está atrelada a nenhum método em específico, mas sim ao problema, e pode ser adotada em qualquer circunstância na qual faça sentido uma inicialização aperfeiçoada da matriz de posto baixo.

---

**Algoritmo 4 – Determinação do Ponto Inicial para o Problema (2.5)**


---

```

1 entrada: Matriz de dados  $A \in \mathbb{R}^{m \times n}$  não nula, seus valores singulares
    $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p > 0$ , em que  $p \leq \min\{m, n\}$ , assim como os
   respectivos  $2p$  vetores singulares: à esquerda,  $u_1, u_2, \dots, u_p \in \mathbb{R}^m$  e, à
   direita,  $v_1, v_2, \dots, v_p \in \mathbb{R}^n$ ; parâmetro de compromisso  $t \in (0, 1)$ 
   associado ao Problema (2.5); e uma tolerância  $\epsilon_0 \geq 0$  para considerar
   um valor singular como nulo.
2 saída : Componentes matriciais  $L^{(0)}$  e  $S^{(0)}$  do ponto inicial.
3 % tentativa inicial:  $L^{(0)}$  é matriz nula e  $S^{(0)}$  é  $A$ 
4  $L^{(0)} \leftarrow 0_{m \times n}$ ;
5  $f \leftarrow t \|A\|_1$ ;
6 % inicializações auxiliares
7  $L_{trial} \leftarrow 0_{m \times n}$ ;
8  $k \leftarrow 0$ ;
9 enquanto  $k < p$  e  $\sigma_{(k+1)} > \epsilon_0$  faça
10 | % complementar o ponto tentativo
11 |  $L_{trial} \leftarrow L_{trial} + \sigma_{(k+1)} u_{(k+1)} \cdot v_{(k+1)}^\top$ ;
12 | % calcular valor funcional associado
13 |  $f_{trial} \leftarrow t \|A - L_{trial}\|_1 + (1 - t) \|L_{trial}\|_*$ ;
14 | se  $f_{trial} < f$  então
15 | | % consolidar a tentativa
16 | |  $f \leftarrow f_{trial}$ ;
17 | |  $L^{(0)} \leftarrow L_{trial}$ ;
18 | senão
19 | | retorne  $L^{(0)}$  e  $A - L^{(0)}$ , e termine.
20 | fim
21 | % incrementar contador de posto
22 |  $k \leftarrow k + 1$ ;
23 fim
24 retorne  $L^{(0)}$  e  $A - L^{(0)}$ , e termine.

```

---

As dificuldades dessa estratégia são essencialmente duas. A primeira está no fato de que não sabemos o posto da solução do problema, sequer aproximadamente,

apesar de supormos que seja *pequeno*. A segunda é que, em geral, não temos ideia sobre qual das duas componente matriciais exerce mais influência nos maiores valores singulares de  $A$ . Em experimentos comparando os valores singulares das componentes-solução  $S^*$  e  $L^*$  com os da matriz  $A$ , mostramos que é possível haver diferenças de ordem de grandeza entre os valores de  $\sigma_{S^*}$  e  $\sigma_{L^*}$ , de maneira que os maiores valores de  $\sigma_A$  podem ser “herdados” de uma (e qualquer uma) de suas componentes (ver Seção 4.4). Portanto, a decisão sobre quantos valores singulares de  $A$  considerar – se é que deveríamos incorporar algum – seria arbitrária e fortemente atrelada ao tipo de instância com que estaríamos lidando.

### 8.1.1 Resultados Numéricos da Partida Aprimorada

Realizamos novos experimentos numéricos com as instâncias aleatórias e com as de videovigilância, incorporando a heurística de partida aprimorada do Algoritmo 4 tanto ao SPGM quanto ao ADMM. Para indicar a incorporação desse procedimento a esses métodos, usaremos a notação com o símbolo “+”: ADMM+ e SPGM+, ou, simplesmente, A+ e S+. Os parâmetros das configurações de execução são exibidos nas Tabelas 18 e 19. O posto do ponto inicial proveniente da heurística de partida aprimorada pode ser consultado nas Tabelas 35 e 36, assim como outras de suas características.

No que segue, apresentamos e discutimos os resultados de erro relativo para as instâncias aleatórias e, depois, para as de videovigilância.

Tabela 18 – Parâmetros do estudo de desempenho do SPGM com e sem *warm-start*.

$(S^{(0)}, L^{(0)}) : \text{Alg. 4}$	$\mu \in 10^{\{-4, -3, -2, -1\}}$	$T = 4$	$M_{\text{it}} = 100$	$\sigma = 10^{-4}$
$\epsilon_x = 10^{-6}$	$\alpha_{\min} = 10^{-30} = 1/\alpha_{\max}$	$M = 20$	$M_{\text{itLS}} = 20$	$\zeta = 0.5$

Tabela 19 – Parâmetros do estudo de desempenho do ADMM com e sem *warm-start*

$(S^{(0)}, L^{(0)}) : \text{Alg. 4}$	$\rho = \frac{n^2}{4 \ \text{vec}(A)\ _1}$	$\epsilon_{\text{relinf}} = \epsilon_x = -\text{Inf}$	$M_{\text{it}} = 120$ (aleatórios)
			$M_{\text{it}} = 1000$ (vídeos)

#### 8.1.1.1 Instâncias Aleatórias

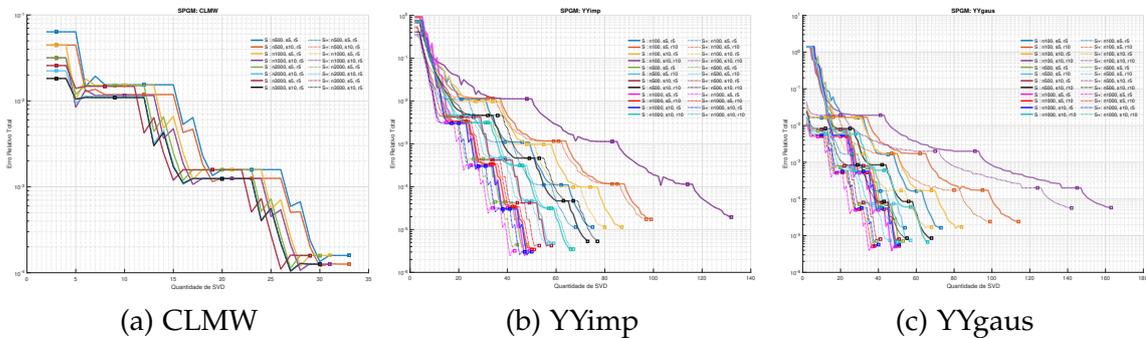
Observamos em nossos experimentos com o *warm-start* que, em todas as instâncias do grupo CAVIAR, o procedimento inicializador resultou em uma matriz nula para a componente atrelada ao posto, isto é,  $L^{(0)} = 0$  e, conseqüentemente,  $S^{(0)} = A$ . Ou seja, não haverá diferenças entre os resultados com ou sem *warm-start* para nenhum dos métodos quando aplicados a esse grupo de instâncias. Tal ocorrência é condizente com a análise estrutural que fizemos na Seção 4.5, na qual avaliamos o erro

relativo associado às aproximações  $A_k$  de  $A$ . Note, porém, que ao adotar o Algoritmo 4 estamos nos apoiando em uma sequência específica de valores para a função objetivo original, e não no erro relativo associado aos candidatos a ponto inicial, como fizemos anteriormente na referida seção. Portanto, no que segue, daremos ênfase apenas às instâncias aleatórias de Yuan e Yang.

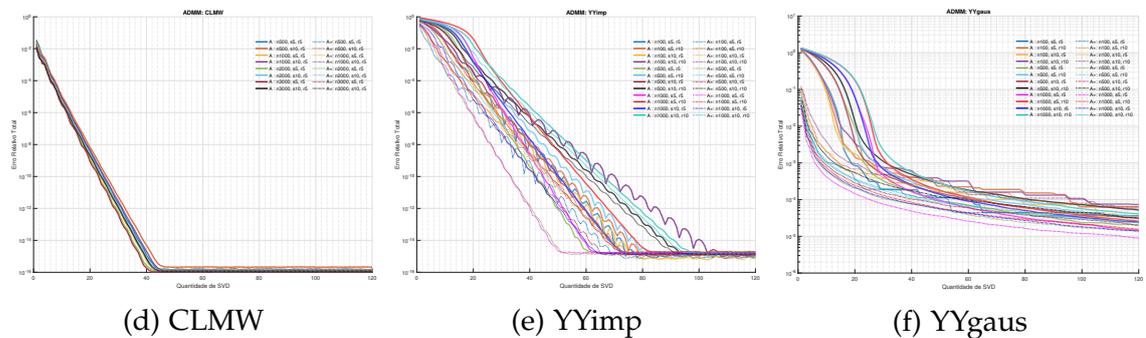
A Figura 23 ilustra os erros relativos obtidos ao longo do processo de otimização com partida aprimorada no que se refere às instâncias aleatórias. Iniciamos fazendo alguns destaques sobre o impacto do *warm-start* nas curvas de erro relativo. Em comum entre o SPGM e o ADMM há o fato de que as curvas se iniciam em pontos do plano mais próximos à origem do que aquelas obtidas antes, sem a inicialização aprimorada, já que o erro inicial agora é menor. Além disso, considerando o nível de precisão em que miramos, com ambos os métodos os efeitos positivos da partida aprimorada são mais notáveis em YYgaus do que em YYimp.

Figura 23 – Comparação de erros relativos ao longo das iterações do ADMM e do SPGM para as instâncias aleatórias, com e sem partida aprimorada (*warm-start*). As configurações experimentais constam nas Tabelas 18 e 19. Os pequenos quadrados nas curvas do SPGM indicam o fim de um ciclo homotópico, ou seja, o término da resolução de um subproblema, durante o qual o parâmetro de suavização é mantido fixo.

SPGM com (S+) e sem (S) *warm-start*



ADMM com (A+) e sem (A) *warm-start*



No caso específico do SPGM, tanto com YYimp (Subfig. 23(b)) quanto com YYgaus (Subfig. 23(c)), parece ocorrer em cada curva um leve encolhimento horizontal

unilateral, da direita para a esquerda, curiosamente quase sem afetar os padrões de oscilação e curvatura. Isto pode sugerir que o ponto inicial aprimorado conduz o SPGM a uma sequência de iterandos muito semelhante àquela percorrida sem esse recurso, porém saltando os elementos iniciais, o que resulta numa aceleração de convergência cuja intensidade depende do tamanho desse salto. Por outro lado, no caso do ADMM, esse fenômeno não é observado. Mesmo com o grupo  $YY_{imp}$  (Subfig. 23(e)), em que certos comportamentos oscilatórios se repetem com ou sem *warm-start*, estes parecem ser mais característicos da alternância de direções do ADMM do que da sequência de pontos percorrida, uma vez que há discrepâncias visíveis nos padrões de oscilação. Por fim, o maior impacto do *warm-start* na mudança das curvaturas das curvas de erro ocorre com o ADMM no conjunto  $YY_{gaus}$ , em que observamos uma mudança de concavidade decorrente de uma notável aceleração do método nas primeiras iterações com a partida aprimorada. De acordo com nossa análise no Capítulo 6 a respeito das estruturas das instâncias, segundo a qual os vídeos são mais semelhantes às instâncias gaussianas do que às impulsivas, o comportamento que acabamos de descrever também sugere que o *warm-start* impactará positivamente os resultados com as instâncias de videovigilância.

Outras particularidades de cada método podem ser comentadas. Pensando numa escala de dificuldades em resolver cada uma das instâncias aleatórias, e baseados no erro relativo, vemos que a adoção do *warm-start* não afetou essa escala no caso do SPGM; em particular, nas instâncias de Yuan e Yang, os quatro problemas mais difíceis continuaram sendo os de menor dimensão,  $n = 100$ , como já havíamos notado na Seção 4.6. Isso corrobora nossa suspeita de que o SPGM consegue melhores desempenhos em instâncias de maior porte do que  $n = 100$ , o que conseqüentemente acaba desfavorecendo o método quando adotamos conjuntos de teste que contenham dimensões semelhantes a essa. Já com o ADMM, houve alterações em sua escala de dificuldades, notavelmente no grupo  $YY_{gaus}$ , como constatamos ao ver que as curvas mudaram suas posições relativas na comparação com e sem partida aprimorada, na Subfigura 23(f).

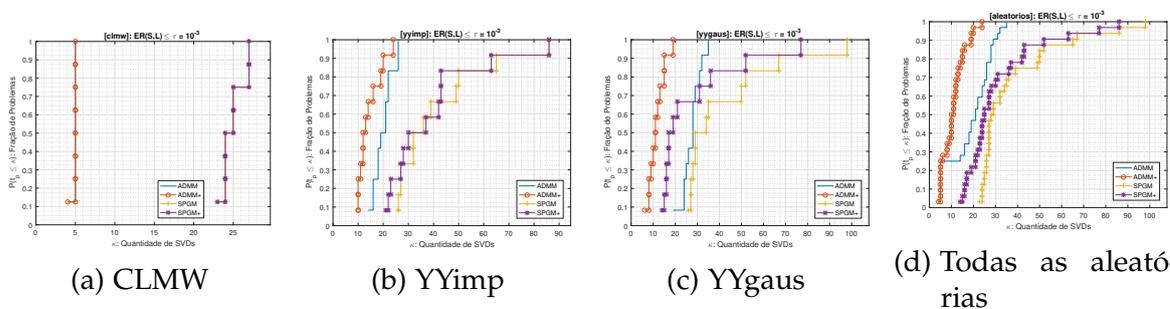
Ainda a respeito do efeito do *warm-start* sobre o ADMM nas instâncias gaussianas, em que pese a aceleração da convergência nas iterações iniciais, os resultados sugerem que há uma precisão limite que o método consegue atingir nessa classe de problemas, já que as curvas pontilhadas e contínuas tendem a se encontrar após algumas centenas de iterações, na Subfigura 23(f). Algo semelhante também ocorre na Subfigura 23(e), no caso impulsivo, com a diferença que, neste último caso, os erros atingidos podem ser considerados *numericamente nulos*.

De um modo geral, a adoção da heurística de *warm-start* acarretou, em quase todos os casos, precisões tão boas ou melhores do que as obtidas com partida

trivial nas respectivas instâncias, não apenas ao final do processo mas também a cada SVD realizada. De fato, se suprimimos o grupo CLMW e se consideramos os erros como função dos gastos, constatamos que, em quase todas as instâncias, os erros com *warm-start* são pontualmente menores do que aqueles obtidos sem essa heurística, o que visualmente se traduz em curvas pontilhadas abaixo das linhas sólidas correspondentes. A título de exemplo, duas das raras exceções são as instâncias de características  $(n, s, r) = (500, 10, 10)$  de YYimp no caso do SPGM (Subfig. 23(b)) e  $(n, s, r) = (1000, 10, 10)$  de YYgaus no caso do ADMM, neste último apenas na vizinhança da iteração 100 (Subfig. 23(f)). Sendo assim, os indícios são amplamente favoráveis ao uso da heurística.

A fim de facilitar a comparação direta de desempenhos do SPGM e do ADMM nas instâncias aleatórias com e sem *warm-start*, condensamos os resultados que acabamos de apresentar nos *data profiles* contidos na Figura 24. A Tabela 20, análoga à Tabela 17, classifica os problemas como resolvidos ou não, novamente de acordo com o teste de convergência (10). O que se destaca de diferente da situação sem a partida aprimorada, discutidas na Seção 4.6 e no Capítulo 7, é o impulsionamento do ADMM nas instâncias gaussianas, justamente aquelas em que o SPGM era comparativamente mais promissor e que o levaram a superar o método de primeira ordem nas instâncias de vídeos (cf. Cap. 7).

Figura 24 – Comparação da quantidade de problemas resolvidos, dentre as instâncias aleatórias, em termos dos gastos efetuados para resolvê-los, com e sem partida aprimorada (*warm-start*), cuja realização é indicada por “+”. As curvas indicam o percentual de problemas que cada *solver* consegue resolver, de acordo com o critério que considera a instância resolvida quando o erro relativo atingido é menor ou igual a  $10^{-3}$ . Neste trabalho, essa precisão foi adotada como alvo com o objetivo de gerar imagens em que a separação do plano de fundo é realizada com qualidade visual tão favorável quanto permite a abordagem adotada para o problema convexo original.



Apesar disso, é preciso manter sempre em mente três aspectos já discutidos. O primeiro, identificado por meio das instâncias de CLMW na Seção 4.6, é que, pra aproveitar um ponto inicial bom, o SPGM precisa ter sua configuração de parâmetros

adaptada, em particular o valor de  $\mu^{(0)}$ . Neste trabalho não nos debruçamos sobre essa calibração, embora seja um aspecto que mereça futuras investigações tendo em vista seu possível impacto para um bom desempenho do método. O segundo aspecto é o aparente desfavorecimento do SPGM nas instâncias menores dos conjuntos de testes. Por fim, há também os indícios de que a estrutura das instâncias impulsivas, em particular as do grupo CLMW, não encontram respaldo nas situações de aplicação que requerem o uso da abordagem SLRMD. Veremos, a seguir, como a partida aprimorada interferiu nos resultados com as instâncias de videovigilância.

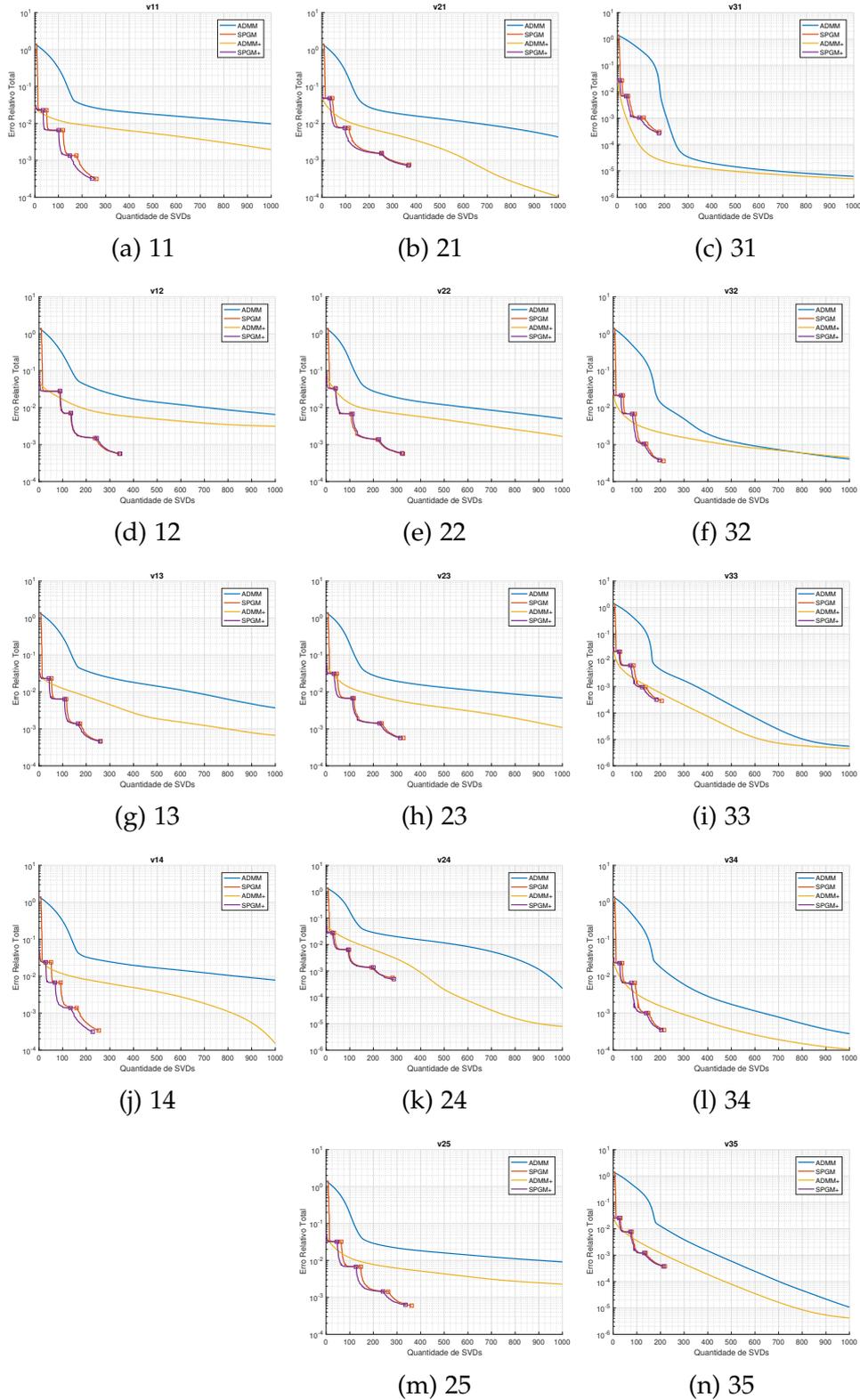
#### 8.1.1.2 Instâncias de Videovigilância

Por fim, direcionamos as atenções aos efeitos do procedimento de *warm-start* no desempenho dos métodos SPGM e ADMM na resolução dos problemas práticos de videovigilância. Os experimentos com ambos resolvidores foram executados seguindo as configurações das Tabelas 18 e 19. Os gráficos de erro relativo em termos dos gastos em número de SVDs se encontram na Figura 25, para as instâncias do grupo CAVIAR, e na Figura 26, para o grupo SBI. Assim como antes, condensamos os resultados por meio de *data profiles* e os exibimos na Figura 27. Para fins de consulta, a especificação das instâncias resolvidas, segundo o teste de convergência que vimos adotando, é feita na Tabela 20, que é análoga à Tabela 17. Além disso, o posto da componente  $L^{(0)}$  do ponto inicial, assim como outras de suas características, podem ser acompanhados nas 35 e Tabelas 36. Adicionalmente, para fins de consulta e completude, disponibilizamos os gráficos das medidas de ER, PEP e EMPE com *warm-start* no Apêndice I.

Optamos por forçar o ADMM a realizar mil iterações, com o objetivo de garantir que sua execução não seria interrompida com orçamento inferior àquele do SPGM, tornando, assim, a comparação mais transparente. Essa decisão fez com que as curvas do ADMM se prolonguem muito mais do que as do SPGM e, nesse sentido, vale ter em mente que o SPGM está configurado para ambicionar erros relativos de 0.1%, de forma que seu progresso e sua parada são influenciados por esse nível de precisão predeterminado. Em quatro instâncias do SBI, ocorreu de a solução de referência ser recuperada pelo ADMM antes do término das mil iterações, o que graficamente pode ser identificado pelos bicos indicando erros numericamente desprezíveis, seguidos de um crescimento no erro retratado. São elas: CAVIAR1, CAVIAR2, HighwayI e HighwayII. Esses fenômenos não impactam as análises que fazemos aqui.

Os resultados mostram que, em geral, a heurística de determinação do ponto inicial melhorou ou manteve o desempenho de ambos os métodos em todas as instâncias práticas. O que mais se destaca na comparação é o fato de que o *warm-start* melhorou substancialmente o desempenho do ADMM nos níveis de precisão em que estamos interessados, enquanto, por outro lado, o progresso no SPGM foi bastante

Figura 25 – Comparação de erros relativos ao longo das iterações do ADMM e do SPGM para as instâncias do Grupo CAVIAR, com e sem partida aprimorada (*warm-start*). As configurações experimentais constam nas Tabelas 18 e 19. Os pequenos quadrados nas curvas do SPGM indicam o fim de um ciclo homotópico, ou seja, o término da resolução de um subproblema, durante o qual o parâmetro de suavização é mantido fixado.



(a) 11

(b) 21

(c) 31

(d) 12

(e) 22

(f) 32

(g) 13

(h) 23

(i) 33

(j) 14

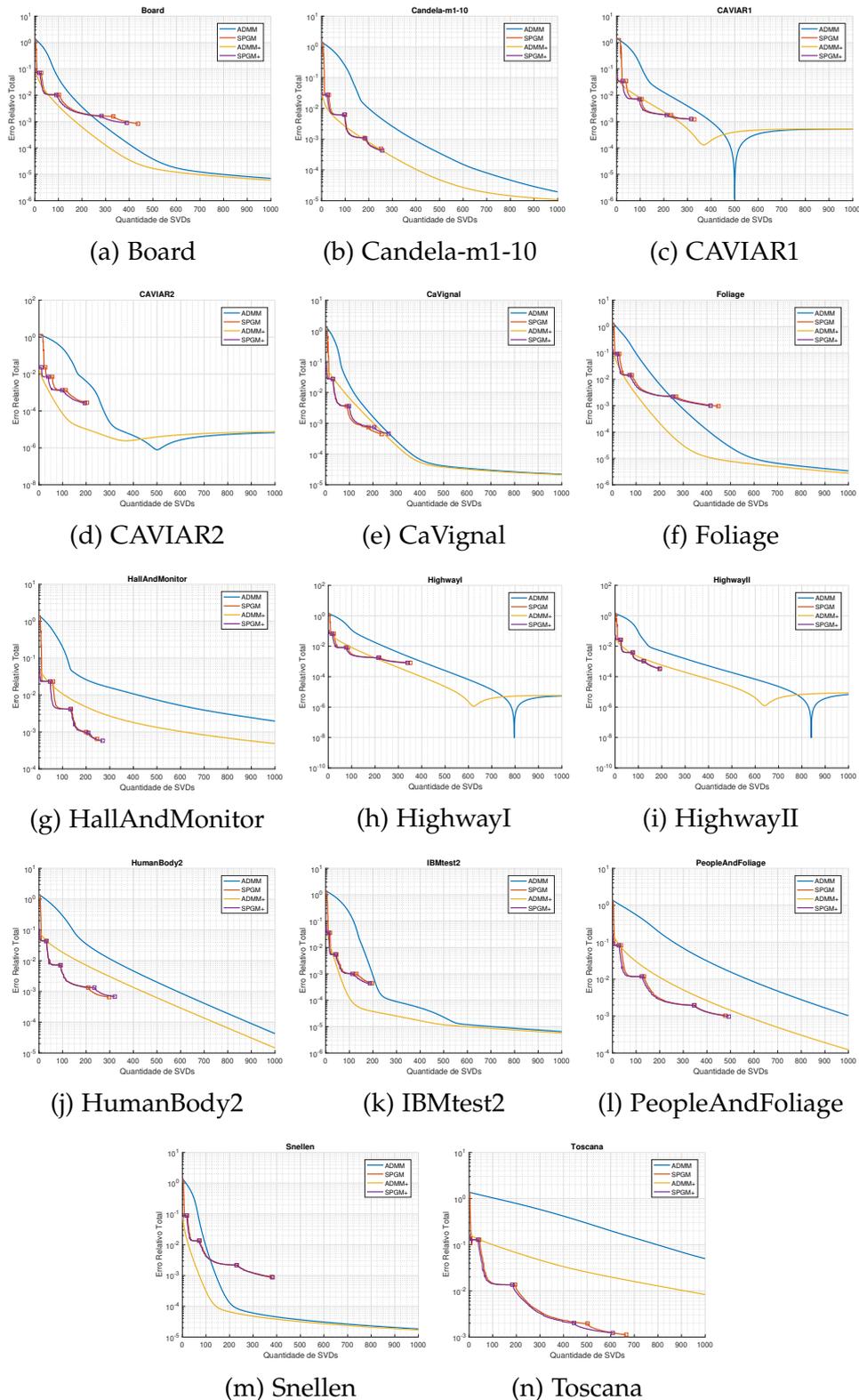
(k) 24

(l) 34

(m) 25

(n) 35

Figura 26 – Comparação de erros relativos ao longo das iterações do ADMM e do SPGM para as instâncias do Grupo SBI, com e sem partida aprimorada (*warm-start*). As configurações experimentais constam nas Tabelas 18 e 19. Os pequenos quadrados nas curvas do SPGM indicam o fim de um ciclo homotópico, ou seja, o término da resolução de um subproblema, durante o qual o parâmetro de suavização é mantido fixo.



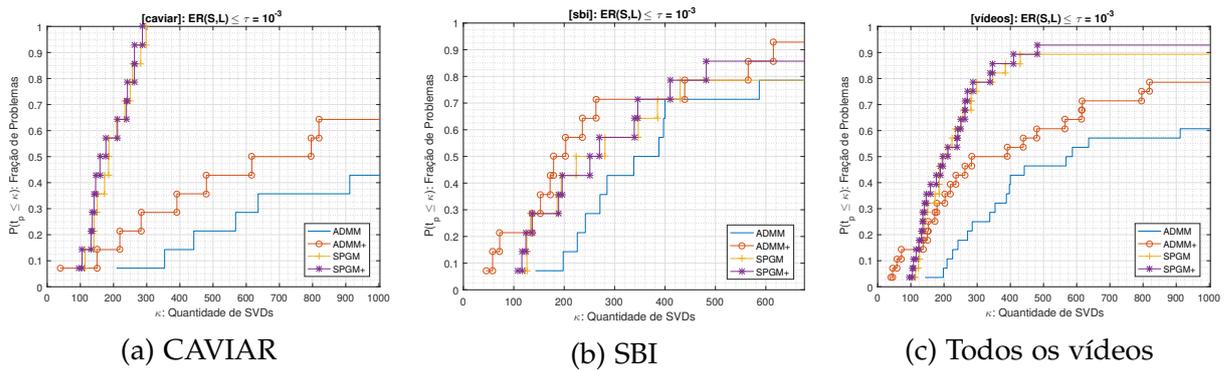
tímido. Ainda assim, a rigor, uma instância a mais foi decretada como resolvida pelo SPGM, em contraposição ao uso do ponto inicial trivial: o problema *PeopleAndFoliage*. A evolução para o ADMM foi muito além: o método passou a resolver os problemas 13, 14 e 21, do grupo *CAVIAR*, e *HallAndMonitor* e *PeopleAndFoliage*, do grupo *SBI*, os quais não solucionava antes (cf. Tabelas 17 e 20). Considerando os dois grupos, o SPGM foi capaz de resolver com o aprimoramento do ponto inicial o total de 26 problemas, ao passo que o ADMM somou 22 êxitos.

Para além da precisão que cada método consegue atingir ao *final* do processo de minimização, reforçamos que num contexto de aplicação é fundamental analisar o que se verifica *no decorrer* de sua execução, já que os usuários da otimização tipicamente têm limitações orçamentárias diversas. Nesse sentido, os *data profiles* da Figura 27 e seus degraus fornecem informações importantes. Dentre as instâncias do grupo *CAVIAR*, o SPGM+ resolve com mais eficiência que o ADMM+ mais de 90% dos problemas. Nesse mesmo grupo, a propósito, enquanto o SPGM+ se mostrou mais robusto ao solucionar todas as instâncias, o ADMM+ resolveu menos do que 65% delas. Já no grupo *SBI*, a discrepância foi menos acentuada e as curvas dos *profiles* de ambos os métodos com partida aprimorada se mantiveram próximas e até se entrelaçaram em alguns momentos. O ADMM+ foi mais eficiente em aproximadamente 64% dos problemas, enquanto para o SPGM+ esse número foi de pouco mais de 14%. Em robustez, o ADMM+ se saiu melhor resolvendo um problema a mais do que o SPGM+, o qual alcançou o objetivo em 86% dos problemas. Por fim, ao considerar todos os vídeos, vemos na Subfigura 27(c) que o peso do grupo *CAVIAR* se sobrepõe ao do *SBI*, tornando o SPGM+ mais eficiente e mais robusto do que o ADMM+ nesse conjunto.

A conclusão é que o *warm-start* melhora significativamente o desempenho do ADMM a ponto de fazê-lo superar o SPGM em várias instâncias, particularmente nas do grupo *SBI*, tanto em termos de eficiência quanto em robustez. Ainda assim, o SPGM sustenta seu bom desempenho e, na comparação geral dos vídeos, suplantou o ADMM na maioria dos casos.

Um questionamento que remanesce diz respeito à existência de uma configuração de parâmetros e estratégia de homotopia que façam o SPGM aproveitar melhor o *warm-start* à disposição. Vimos nas experiências com as instâncias aleatórias de *CLMW*, quando ainda estávamos por fixar uma configuração, na Seção 4.6, que o fato do ponto inicial trivial já estar por si só associado a erros relativos comparativamente *pequenos*, a diminuição de  $\mu_0$  traria avanços consideráveis ao desempenho do SPGM. No caso do *warm-start*, em que também incorremos em erros comparativamente *pequenos*, podemos conjecturar que outras configurações para o SPGM poderiam aproveitar melhor a partida aprimorada. Em experimentos preliminares, no entanto, a simples modificação de  $\mu_0$  não surtiu tanto efeito, de forma que acreditamos que todos os outros parâmetros

Figura 27 – Comparação da quantidade de problemas resolvidos, dentre as instâncias dos vídeos, em termos dos gastos efetuados para resolvê-los, com e sem partida aprimorada (*warm-start*), cuja realização é indicada por “+”. As curvas indicam o percentual de problemas que cada *solver* consegue resolver, de acordo com o critério que considera a instância resolvida quando o erro relativo atingido é menor ou igual a  $10^{-3}$ . Neste trabalho, essa precisão foi adotada como alvo com o objetivo de gerar imagens em que a separação do plano de fundo é realizada com qualidade visual tão favorável quanto permite a abordagem adotada para o problema convexo original.



precisem ser repensados, atualizados e testados – um caminho para investigações futuras.

Tabela 20 – Indicação de quais problemas cada *solver* foi capaz de resolver com erro relativo menor do que ou igual a  $1.0 \times 10^{-3}$ , seguindo os critérios de parada apresentados, usando o *warm-start* para o ponto inicial. O símbolo  $\checkmark$  indica que a precisão alvo foi atingida, enquanto  $\times$  expressa o contrário. O “+” indica *warm-start*.

Instância	SPGM+	ADMM+	Instância	SPGM+	ADMM+
11	$\checkmark$	$\times$	Board	$\checkmark$	$\checkmark$
21	$\checkmark$	$\checkmark$	Candela_m1.10	$\checkmark$	$\checkmark$
31	$\checkmark$	$\checkmark$	CAVIAR1	$\times$	$\checkmark$
12	$\checkmark$	$\times$	CAVIAR2	$\checkmark$	$\checkmark$
22	$\checkmark$	$\times$	CaVignal	$\checkmark$	$\checkmark$
32	$\checkmark$	$\checkmark$	Foliage	$\checkmark$	$\checkmark$
13	$\checkmark$	$\checkmark$	HallAndMonitor	$\checkmark$	$\checkmark$
23	$\checkmark$	$\times$	HighwayI	$\checkmark$	$\checkmark$
33	$\checkmark$	$\checkmark$	HighwayII	$\checkmark$	$\checkmark$
14	$\checkmark$	$\checkmark$	HumanBody2	$\checkmark$	$\checkmark$
24	$\checkmark$	$\checkmark$	IBMtest2	$\checkmark$	$\checkmark$
34	$\checkmark$	$\checkmark$	PeopleAndFoliage	$\checkmark$	$\checkmark$
25	$\checkmark$	$\times$	Snellen	$\checkmark$	$\checkmark$
35	$\checkmark$	$\checkmark$	Toscana	$\times$	$\times$
SUCESSOS:	14	9	SUCESSOS:	12	13

## 8.2 Pontos Iniciais no Espaço de Exibição

Na Seção anterior, finalizamos a análise do desempenho dos métodos SPGM e ADMM na resolução de problemas de otimização do tipo SLRMD adotando instâncias aleatórias e práticas, com e sem o uso do Algoritmo 4 de determinação do ponto inicial. Nesta seção, discutimos alguns aspectos visuais dos pontos oriundos dessa heurística, comparando-os às soluções de referência e às matrizes de posto unitário que melhor aproximam os dados de entrada em cada problema.

Até o momento, vimos que a estratégia de associar  $L$  às melhores aproximações de  $A$  não constitui uma maneira consistente e eficaz de resolver a formulação convexa do problema de decomposição posto-esparsidade (Problema (2.5)). Em outras palavras, os pontos obtidos não minimizam a função objetivo do problema restrito. Uma constatação relacionada, porém distinta, é a de que tais pontos, em geral, não possuem as características de esparsidade e o posto baixo como almejado, objetivo que em si precede a formulação do problema de otimização. Apesar disso, vimos que o uso dessa estratégia na construção de uma heurística de ponto inicial é promissor na aceleração dos métodos de otimização, especialmente do ADMM. Vimos também que, na prática, obter valores abaixo de 0.1% para o erro relativo à solução de referência é suficiente para que a imagem correspondente seja bastante próxima daquela associada a essa solução – que é a melhor solução que temos para o problema, no espaço de otimização – nos termos das medidas no espaço de exibição.

Os questionamentos que suscitamos agora se referem ao que pode ser observado no espaço de exibição e visam confrontar, segundo as expectativas da aplicação, alguns aspectos visuais de três tipos de pontos: das soluções de referência, dos pontos iniciais da heurística de *warm-start* e das melhores aproximações de posto unitário para as matrizes de dados de entrada. Observe que a análise da precisão suficiente diz respeito ao problema de otimização, enquanto que a visualização desses pontos diz respeito ao que antecede a formulação matemática: a aplicação da separação do plano de fundo. A quais imagens no espaço de exibição correspondem os *warm-starts* dos vídeos? São imagens comparáveis às soluções de referência? E a aproximação de posto unitário para  $A$ , econômica por construção, se assemelha visualmente a quê? Como são essas matrizes no espaço de exibição? Para tentar responder a essas questões, reconstruímos as imagens a partir dos dados numéricos.

### Comparação de Imagens

As Figuras 28 e 29 exibem uma seleção de *frames* das soluções de referência, dos pontos de *warm-start* e das aproximações de posto 1. Os *frames* estão dispostos em cada item da figura em blocos de  $3 \times 2$  imagens. As lacunas na última linha indicam

que haveria repetição da imagem de cima por motivo de igualdade entre as matrizes advindas do *warm-start* e as da aproximação de posto 1. Vale lembrar que cada *frame* corresponde a uma coluna da respectiva matriz do problema de otimização e que as entradas eventualmente negativas têm o sinal invertido para fins de exibição (cf. Seção 5.2). A escolha dos instantes a serem retratados em cada instância foi baseada meramente na ocorrência de objetos em movimento na cena original, a partir de uma pré-seleção visual com base na inspeção de uma dezena de imagens igualmente espaçadas no tempo.

Em que pese a limitação do olho humano na comparação visual, há diferenças praticamente imperceptíveis entre as imagens correspondentes nas instâncias de v11, v31, v32 e v34, do grupo CAVIAR, e CAVIAR2, do grupo SBI; nas demais, as discrepâncias são mais facilmente percebidas. Em comum entre as instâncias listadas está o fato de que, em todas essas cinco, o *warm-start* e a aproximação de posto 1 coincidiram, o que significa que o posto unitário levou vantagem com relação aos demais na minimização da função objetivo a partir da SVD de  $A$ . Note, no entanto, que há outras três instâncias em que o *warm-start* e a aproximação de posto 1 coincidiram sem que as imagens fossem tão semelhantes (v13, IBMtest2 e Toscana), o que dificulta a análise causal. É importante ter em mente que as imagens exibidas se referem a apenas um dos vários *frames* das cenas, o que poderia nos levar a generalizações equivocadas. Por isso, nosso foco aqui está nas que apresentam diferenças mais significativas.

De um modo geral, visualmente, os *frames* do *warm-start* podem ser interpretadas como aproximações intermediárias entre as imagens das soluções de referência e as das aproximações de posto unitário, já que, a rigor, a matriz do *warm-start* tem como uma de suas parcelas constituintes justamente a aproximação  $A_1$ .

À primeira vista, chama à atenção uma aparente qualidade dos *frames* oriundos da aproximação de posto 1, os quais são obtidos sem a execução de métodos de otimização e, portanto, sem custos. Tal estratégia conseguiu produzir *frames* para o plano de fundo que recuperaram algo semelhante, a olho nu, ao cenário original, ainda que com borrões dos objetos em movimento ou com comprometimento das intensidades, perdidas para a componente que seria de esparsidade. Para algumas instâncias, como as que mencionamos há pouco como indistintas, a estratégia surpreendentemente parece tão eficiente quanto o uso da otimização. Porém, uma observação mais minuciosa mostra que, para além do que já concluímos na Seção 4.5 sobre essas aproximações, não se trata nem de resultados robustos nem de qualidade no que diz respeito aos objetivos de decomposição posto-esparsidade.

A aparente eficiência da matriz de posto unitário em obter uma decomposição a custo quase nulo se trata mais de uma eventualidade por particularidades das instâncias do que uma maneira sistemática de separação. De fato, basta ver que é fre-

Figura 28 – Seleção de *frames* das soluções de referência, dos pontos de *warm-start* e das aproximações de posto 1 das instâncias do grupo CAVIAR. Em cada bloco de imagens referentes a uma mesma instância, de cima para baixo, as linhas se referem à solução de referência, ao ponto de *warm-start* e, quando houver diferença, à aproximação de posto 1; a coluna da esquerda corresponde à componente do posto e a da direita, à da esparsidade. Instâncias e *frames* indicados.

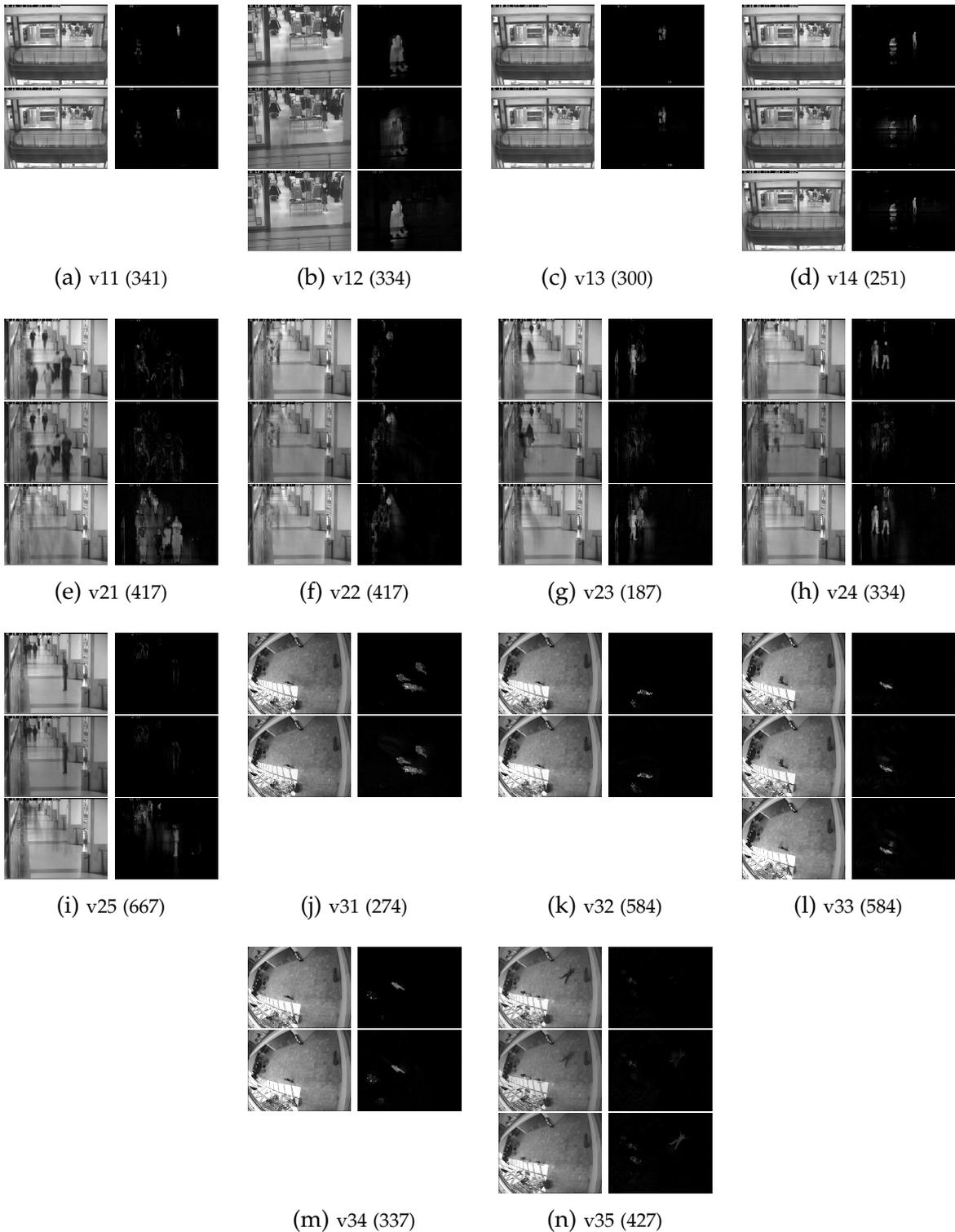
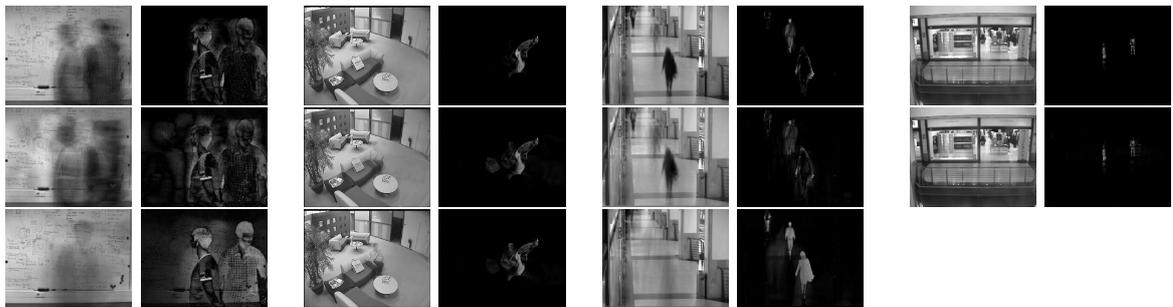


Figura 29 – Seleção de *frames* das soluções de referência, dos pontos de *warm-start* e das aproximações de posto 1 das instâncias do grupo SBI. Em cada bloco de imagens referentes a uma mesma instância, de cima para baixo, as linhas se referem à solução de referência, ao ponto de *warm-start* e, quando houver diferença, à aproximação de posto 1; a coluna da esquerda corresponde à componente do posto e a da direita, à da esparsidade. Instâncias e *frames* indicados.



(a) Board (203)

(b) Candela (311)

(c) CAVIAR1 (475)

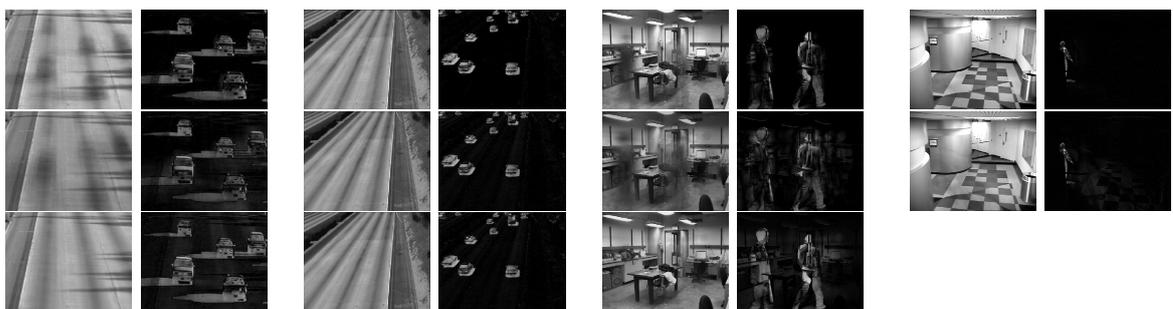
(d) CAVIAR2 (154)



(e) CaVignal (144)

(f) Foliage (176)

(g) Hall&Monitor (132)

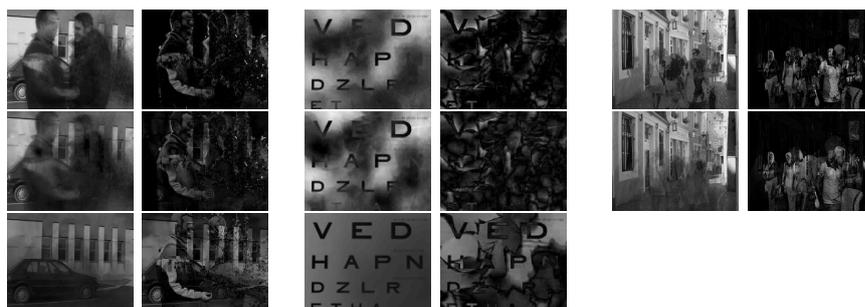


(h) HighwayI (294)

(i) HighwayII (56)

(j) HumanBody2 (576)

(k) IBMtest2 (1)



(l) PeopleAndFolige (77)

(m) Snellen (250)

(n) Toscana (6)

quente a aparição de parte do plano de fundo na componente designada para capturar os movimentos, como, por exemplo, nas instâncias v21 e HumanBody2. Para além da aparente separação visual, as componentes matriciais correspondentes a essas imagens não são esparsas ou possuem posto baixo, nem mesmo sob grandes tolerâncias.

Por outro lado, as imagens das soluções de referência nem sempre escapam de estabelecer como modelos imagens que nitidamente não correspondem à decomposição desejada, como podemos observar nas imagens de v21 e Snellen, por exemplo. Uma provável explicação para esses casos está na falta de garantia teórica de que a decomposição é possível de ser feita através do problema SLRMD, uma vez que saímos do universo controlado das instâncias aleatórias. Ainda que houvesse tal garantia, poderia ser o caso de que ela estivesse atrelada a faixas desconhecidas de valores para o parâmetro  $t$  de compromisso entre o posto e a esparsidade na função objetivo. Vale lembrar que, neste trabalho, herdamos um valor de  $t$  pensado para garantir teoricamente a recuperabilidade nas instâncias aleatórias, e não nas práticas.

Apesar disso, as soluções de referência mostram especial consistência na imagem da componente matricial de posto baixo. Os contornos dos objetos móveis estão mais bem delineados do que nas aproximações de posto 1. Adicionalmente, mesmo os borrões que aparecem nas soluções de referência tendem a ser exclusivos daquele instante que retratam, diferentemente das outras estratégias, que podem herdar borrões de outros momentos da gravação, como constatamos ao ver “duas” pessoas na componente  $S$  de Candela. Com relação às imagens de plano de fundo, as soluções de referência, por tipicamente terem posto muito maior do que 1, permitem a incorporação à componente  $L$  de objetos que passam a ser estáticos em algum instante do decorrer da cena. Como exemplos, temos em v33 uma caixa que é abandonada no chão durante a gravação; em v35, o homem caído no chão após uma briga; em Candela, a sacola esquecida no sofá; em Hall&Monitor, a maleta deixada no banco do corredor; dentre outros.

Evidentemente, os pormenores do comportamento desejado para a decomposição podem variar com a intenção da aplicação (se o alvo é obter o plano de fundo, identificar os objetos móveis, incorporar ao plano de fundo o que deixa de se mexer durante a cena etc.). No entanto, ao tratarmos do problema posto-esparsidade, não é razoável considerar como viáveis estratégias que, em geral, resultam em componentes de posto completo e densas.

## Parte IV

### Considerações Finais

## 9 CONSIDERAÇÕES FINAIS

Neste trabalho, investigamos o problema de decompor uma dada matriz como a soma de uma componente esparsa com outra de posto baixo. Constatamos que, sem hipóteses adicionais a essas, o problema pode ser insolúvel ou até mesmo ter soluções múltiplas. As garantias teóricas que invocamos requerem a ocorrência de certos padrões para que esteja garantida a recuperabilidade e a unicidade da solução do problema.

Exploramos algumas formulações para o problema com diferentes características – favoráveis ou não – desde a primeira formulação contendo explicitamente as funções posto e esparsidade, intratáveis sob o ponto de vista da otimização, até as sucessivas aproximações por modelos quadráticos convenientemente construídos para terem solução fechada.

Em nossos experimentos computacionais, lidamos majoritariamente bem com instâncias de dezenas de milhões de variáveis, vindas de vídeos reais. Constatamos discrepâncias entre as estruturas das matrizes sintéticas e aquelas decorrentes de filmagens reais, e observamos que essas diferenças têm impacto não apenas no desempenho dos métodos mas também nas conclusões que tiramos a partir deles. Nessa mesma direção, apresentamos as evidências de que a busca por soluções de alta precisão podem não se justificar no contexto da aplicação e que é relevante analisar os resultados em função dos gastos despendidos, e não apenas ao final do processo.

Desenvolvemos um processo simples e universal de escolher o ponto inicial do problema a custos ínfimos, e constatamos que seu efeito pode ser decisivo para alguns métodos, ao passo que não se mostrou prejudicial nem mesmo no caso em que as configurações do método lhe eram desfavoráveis.

Por fim, propusemos um método de resolução numérica para o problema matricial que combina estratégias preexistentes de maneira inovadora, até onde é de nosso conhecimento, e que se mostrou eficiente e robusto em diversos cenários. Com isto, qualificamos a continuação homotópica sobre o valor do parâmetro de suavização, atrelada às demais especificações, como promissora no estudo de outras aplicações.

### Dificuldades

Com relação à continuação homotópica, encontramos dificuldades em estabelecer uma configuração de parâmetros que contemplasse diferentes tipos de instância de teste, entre aleatórias e de vídeos, e que extraísse do método as potencialidades

corroborados por nossos resultados. A escassez de estudos sobre o tema, principalmente na mesma área de aplicação, dificulta consideravelmente a escolha dos valores, já que as possibilidades combinam-se e tornam-se numerosas com facilidade. Outra dificuldade foi com o parâmetro de contrapeso entre as normas na função objetivo. Embora tenhamos adotado um valor advindo da literatura, as exibições de algumas das soluções de referência sugerem que o valor padrão possivelmente não é o mais apropriado.

Com relação à filosofia dos métodos computacionais, sob o ponto de vista da aplicação, os desafios ainda são os de possibilitar a entrada constante de *frames* no modelo, sem necessidade dos quadros serem processados por blocos. Outra característica a ser explorada é a relação espacial entre os píxeis, já que sabemos que essas unidades de imagens não são independentes como supõe, na prática, o modelo que adotamos.

## Trabalhos futuros

São boas as perspectivas da suavização da norma nuclear, em conjunto com a escolha adequada da precisão de que se necessita nos casos práticos. É possível explorar a suavização de um ou ambos os termos da função objetivo e, assim, ampliar as possibilidades de ferramentas a serem utilizadas.

Pode ser proveitoso futuramente incorporar ao SPGM certas dinâmicas já existentes, como a da homotopia no parâmetro de contrapeso, a combinação dos dois últimos iterandos no cálculo do ponto seguinte, com o coeficiente apropriadamente escolhido.

## REFERÊNCIAS

- [1] Y. ALTMAN, *export\_fig*, 2021. Disponível em: [https://github.com/altmany/export\\_fig/releases](https://github.com/altmany/export_fig/releases). Acessado em 16/12/2021. (Citado na página 208.)
- [2] N. S. AYBAT, D. GOLDFARB, AND S. MA, *Efficient algorithms for robust and stable principal component pursuit problems*, *Computational Optimization and Applications*, 58 (2014), pp. 1–29. (Citado nas páginas 23 e 25.)
- [3] J. BARZILAI AND J. M. BORWEIN, *Two-point step size gradient methods*, *IMA Journal of Numerical Analysis*, 8 (1988), pp. 141–148. (Citado nas páginas 24, 57 e 59.)
- [4] S. BASU, X. LI, AND G. MICHAILIDIS, *Low rank and structured modeling of high-dimensional vector autoregressions*, *IEEE Transactions on Signal Processing*, 67 (2019), pp. 1207–1222. (Citado na página 19.)
- [5] A. BECK AND M. TEOULLE, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, *SIAM J. Imaging Sci.*, 2 (2009), pp. 183–202. (Citado nas páginas 21, 22, 25 e 63.)
- [6] D. BERTSEKAS, *Convex optimization theory*, Athena Scientific, 2009. (Citado nas páginas 33, 36 e 37.)
- [7] D. P. BERTSEKAS, *Nonlinear Programming*, Athena Scientific, Belmont, 2 ed., 1999. (Citado na página 47.)
- [8] D. P. BERTSEKAS, A. NEDIC, AND A. E. OZDAGLAR, *Convex Analysis and Optimization*, Athena Scientific, Belmont, 2003. (Citado na página 36.)
- [9] E. G. BIRGIN AND J. M. MARTÍNEZ, *Practical augmented Lagrangian methods for constrained optimization*, Society for Industrial and Applied Mathematics, 2014. (Citado na página 21.)
- [10] E. G. BIRGIN, J. M. MARTÍNEZ, AND M. RAYDAN, *Nonmonotone spectral projected gradient methods on convex sets*, *SIAM J. Optim.*, 10 (2000), pp. 1196–1211. (Citado na página 57.)
- [11] T. BOUWMANS, N. S. AYBAT, AND E.-H. ZAHZAH, *Handbook of Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing*, CRC Press, Boca Raton, 2016. (Citado na página 19.)

- [12] T. BOUWMANS, L. MADDALENA, AND A. PETROSINO, *Scene background initialization: A taxonomy*, Pattern Recognition Letters, 96 (2017), pp. 3–11. (Citado nas páginas 23, 120, 122 e 140.)
- [13] T. BOUWMANS, A. SOBRAL, S. JAVED, S. K. JUNG, AND E.-H. ZAHZAH, *Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset*, Computer Science Review, 23 (2017), pp. 1–71. (Citado nas páginas 20 e 23.)
- [14] ———, *Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset*, Comput. Sci. Rev., 23 (2017), pp. 1–71. (Citado nas páginas 21 e 25.)
- [15] T. BOUWMANS AND E. H. ZAHZAH, *Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance*, Computer Vision and Image Understanding, 122 (2014), pp. 22–34. (Citado na página 23.)
- [16] S. BOYD, N. PARIKH, E. CHU, B. PELEATO, AND J. ECKSTEIN, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Foundations and Trends® in Machine Learning, 3 (2011), pp. 1–122. (Citado nas páginas 55, 57 e 145.)
- [17] S. BOYD AND L. VANDENBERGHE, *Convex Optimization*, Cambridge University Press, Cambridge, 2004. (Citado nas páginas 35, 45 e 49.)
- [18] S. R. BUSS, *3D Computer Graphics*, Cambridge University Press, Cambridge, 2003. (Citado na página 202.)
- [19] J.-F. CAI, E. J. CANDÈS, AND Z. SHEN, *A singular value thresholding algorithm for matrix completion*, SIAM J. Optim., 20 (2010), pp. 1956–1982. (Citado nas páginas 21, 22, 58 e 61.)
- [20] J.-F. CAI, S. OSHER, AND Z. SHEN, *Linearized Bregman iterations for compressed sensing*, Math. Comp., 78 (2009), pp. 1515–1536. (Citado nas páginas 21 e 22.)
- [21] E. J. CANDÈS, X. LI, Y. MA, AND J. WRIGHT, *Robust Principal Component Analysis?*, Journal of the ACM, 58 (2011), pp. 1–37. (Citado nas páginas 11, 12, 20, 21, 22, 25, 39, 43, 55, 56, 57, 68, 74, 77, 78, 79, 108, 114, 115, 131, 139, 154, 155 e 223.)
- [22] E. J. CANDÈS AND B. RECHT, *Exact matrix completion via convex optimization*, Comput Math, 9 (2009), pp. 717–772. (Citado nas páginas 20, 21 e 22.)
- [23] CAVIAR, *CAVIAR Test Case Scenarios*, 2004. School of Informatics, University of Edinburgh. Coordinator: Robert B. Fisher. EC Funded, IST 2001 37540. Disponível

- vel em: <http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/>. Acessado em: 16/12/2021. (Citado nas páginas 120 e 140.)
- [24] V. CHANDRASEKARAN, P. A. PARRILO, AND A. S. WILLSKY, *Latent variable graphical model selection via convex optimization*, 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton), 40 (2012), pp. 1935–1967. (Citado nas páginas 19 e 39.)
- [25] V. CHANDRASEKARAN, S. SANGHAVI, P. A. PARRILO, AND A. S. WILLSKY, *Rank-sparsity incoherence for matrix decomposition*, SIAM Journal on Optimization, 21 (2011), pp. 572–596. (Citado nas páginas 20, 21, 39, 40, 41, 42, 43, 44 e 74.)
- [26] J. CHEN, S. LIU, AND M. HUANG, *Low-rank and sparse decomposition model for accelerating dynamic MRI reconstruction*, Journal of Healthcare Engineering, 2017 (2017), pp. 1–9. (Citado na página 19.)
- [27] Y. CHEN, F. WANG, J. WAN, AND K. XU, *Sparse and low-rank decomposition of covariance matrix for efficient DOA estimation*, in 2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN), IEEE, may 2017, pp. 957–961. (Citado nas páginas 19 e 25.)
- [28] P. L. COMBETTES AND V. R. WAJS, *Signal recovery by proximal forward-backward splitting*, Multiscale Modeling & Simulation, 4 (2005), pp. 1168–1200. (Citado nas páginas 21, 56 e 60.)
- [29] I. DAUBECHIES, M. DEFRISE, AND C. D. MOL, *An iterative thresholding algorithm for linear inverse problems with a sparsity constraint*, Comm. Pure Appl. Math., 57 (2004), pp. 1413–1457. (Citado na página 22.)
- [30] E. M. DE SÁ, *Exposed faces and duality for symmetric and unitarily invariant norms*, Linear Algebra and its Applications, 197-198 (1994), pp. 429–450. (Citado na página 38.)
- [31] C. ECKART AND G. YOUNG, *The approximation of one matrix by another of lower rank*, Psychometrika, 1 (1936), pp. 211–218. (Citado na página 53.)
- [32] F. FACCHINEI AND J.-S. PANG, *Finite-Dimensional Variational Inequalities and Complementarity Problems*, vol. 2, Springer Ser. Oper. Res., Springer-Verlag New York Inc., Feb. 2003. (Citado nas páginas 22 e 63.)
- [33] K. FAN, *Maximum properties and inequalities for the eigenvalues of completely continuous operators*, Proc Natl Acad Sci USA, 37 (1951), pp. 760–766. (Citado na página 31.)

- [34] K. FAN AND A. J. HOFFMAN, *Some metric inequalities in the space of matrices*, Proceedings of the American Mathematical Society, 6 (1955), pp. 111–116. (Citado na página 53.)
- [35] M. FAZEL, *Matrix rank minimization with applications*, PhD thesis, PhD thesis, Stanford University, 2002. (Citado nas páginas 31 e 38.)
- [36] M. FAZEL, H. HINDI, AND S. P. BOYD, *A rank minimization heuristic with application to minimum order system approximation*, in Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148), vol. 6, IEEE, 2001, pp. 4734–4739. (Citado na página 19.)
- [37] W. FENCHEL, *On conjugate convex functions*, Canadian Journal of Mathematics, 1 (1949), pp. 73–77. (Citado na página 35.)
- [38] M. FORTIN AND R. GLOWINSKI, *Méthodes de lagrangien augmenté: applications à la résolution numérique de problèmes aux limites*, vol. 28, Dunod Paris, 1982. (Citado na página 21.)
- [39] K. FOUNTOULAKIS AND J. GONDZIO, *A second-order method for strongly convex  $\ell_1$ -regularization problems*, Mathematical Programming, 156 (2016), pp. 189–219. (Citado nas páginas 50, 52 e 100.)
- [40] G. H. GOLUB, *Matrix Computations*, J. Hopkins Uni. Press, 2013. (Citado na página 32.)
- [41] M. L. N. GONÇALVES, M. M. ALVES, AND J. G. MELO, *Pointwise and ergodic convergence rates of a variable metric proximal alternating direction method of multipliers*, Journal of Optimization Theory and Applications, 177 (2018), pp. 448–478. (Citado na página 145.)
- [42] R. C. GONZALEZ AND R. E. WOODS, *Digital Image Processing, Global Edition*, Pearson, fourth ed., 2018. (Citado nas páginas 110, 111, 201, 202, 203, 206, 207 e 208.)
- [43] R. C. GONZALEZ, R. E. WOODS, AND S. L. EDDINS, *Digital Image Processing Using Matlab*, Tata McGraw Hill Education, 2 ed., 2010. (Citado nas páginas 203, 206 e 207.)
- [44] M. GRANT AND S. BOYD, *Graph implementations for nonsmooth convex programs*, in Recent Advances in Learning and Control, V. Blondel, S. Boyd, and H. Kimura, eds., Springer London, 2008, pp. 95–110. (Citado na página 21.)
- [45] —, *CVX: Matlab software for disciplined convex programming, version 2.1*. <http://cvxr.com/cvx>. Acessado em: 16/12/2021., Mar. 2014. (Citado nas páginas 21 e 44.)

- [46] L. GRIPPO AND M. SCIANDRONE, *Nonmonotone globalization techniques for the Barzilai-Borwein gradient method*, Computational Optimization and Applications, 23 (2002), pp. 143–169. (Citado na página 60.)
- [47] J. HADAMARD, *Lectures on Cauchy's problem in linear partial differential equations*, Yale University Press, 1923. (Citado na página 31.)
- [48] E. T. HALE, W. YIN, AND Y. ZHANG, *Fixed-point continuation for  $\ell_1$ -minimization: Methodology and convergence*, SIAM J. Optim., 19 (2008), pp. 1107–1130. (Citado na página 21.)
- [49] B. HE, L. Z. LIAO, D. HAN, AND H. YANG, *A new inexact alternating directions method for monotone variational inequalities*, Math. Program., 92 (2002), pp. 103–118. (Citado na página 57.)
- [50] M. HOHENWARTER, M. BORCHERDS, G. ANCSIN, B. BENCZE, M. BLOSSIER, J. ÉLIÁS, K. FRANK, L. GÁL, A. HOFSTÄTTER, F. JORDAN, Z. KONEČNÝ, Z. KOVÁCS, E. LETTNER, S. LIZELFELNER, B. PARISE, C. SOLYOM-GECSE, C. STADLBAUER, AND M. TOMASCHKO, *GeoGebra 5.0.507.0*, oct 2018. <https://www.geogebra.org>. Acessado em: 16/12/2021. (Citado na página 51.)
- [51] R. A. HORN AND C. R. JOHNSON, *Matrix Analysis*, Cambridge University Press, Cambridge, 2 ed., 2012. (Citado nas páginas 31, 35, 44 e 45.)
- [52] H. HOTELLING, *Analysis of a complex of statistical variables into principal components.*, Journal of Educational Psychology, 24 (1933), pp. 417–441. (Citado na página 23.)
- [53] Y. HUANG AND H. LIU, *A Barzilai-Borwein type method for minimizing composite functions*, Numerical Algorithms, 69 (2015), pp. 819–838. (Citado nas páginas 25, 57 e 60.)
- [54] P. J. HUBER, *Robust estimation of a location parameter*, The Annals of Mathematical Statistics, 35 (1964), pp. 73–101. (Citado na página 49.)
- [55] P. J. HUBER AND E. M. RONCHETTI, *Robust Statistics*, WILEY, 2009. (Citado na página 49.)
- [56] A. IZMAILOV AND M. SOLODOV, *Otimização, Volume 1: Condições de Otimalidade, Elementos de Análise Convexa e de Dualidade*, IMPA, Rio de Janeiro, 2014. (Citado nas páginas 34 e 56.)
- [57] I. T. JOLLIFFE, *Principal Component Analysis*, Springer New York, New York, NY, 1986. (Citado na página 23.)

- [58] S. KONTOGIORGIS AND R. R. MEYER, *A variable-penalty alternating directions method for convex optimization*, Math. Program., 83 (1998), pp. 29–53. (Citado na página 57.)
- [59] J. LEE AND B. ZOU, *Optimal rank-sparsity decomposition*, Journal of Global Optimization, 60 (2014), pp. 307–315. (Citado na página 31.)
- [60] A. S. LEWIS, *The mathematics of eigenvalue optimization*, Math. Program., 97 (2003), pp. 155–176. (Citado na página 38.)
- [61] A. S. LEWIS AND H. S. SENDOV, *Nonsmooth analysis of singular values. part I: Theory*, Set-Valued Analysis, 13 (2005), pp. 213–241. (Citado nas páginas 50 e 51.)
- [62] Z. LIN, M. CHEN, AND Y. MA, *The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices*, arXiv preprint arXiv:1009.5055, (2010). (Citado na página 22.)
- [63] Z. LIU AND L. VANDENBERGHE, *Interior-point method for nuclear norm approximation with application to system identification*, SIAM Journal on Matrix Analysis and Applications, 31 (2009), pp. 1235–1256. (Citado na página 48.)
- [64] J. LÖFBERG, *YALMIP: A toolbox for modeling and optimization in MATLAB*, in In Proceedings of the CACSD Conference, Taipei, Taiwan, 2004. Disponível em <https://yalmip.github.io/>. Acessado em 16/12/2021. (Citado na página 44.)
- [65] J. M. LONG, J. Y. CHUN, AND T. K. GAYLORD, *ADMM approach for efficient iterative tomographic deconvolution reconstruction of 3D quantitative phase images*, Appl. Opt., 60 (2021), pp. 8485–8492. (Citado na página 25.)
- [66] S. MA, D. GOLDFARB, AND L. CHEN, *Fixed point and Bregman iterative methods for matrix rank minimization*, Mathematical Programming, 128 (2011), pp. 321–353. (Citado na página 25.)
- [67] L. MADDALENA AND A. PETROSINO, *Towards benchmarking scene background initialization*, New Trends in Image Analysis and Processing – ICIAP 2015 Workshops, Lecture Notes in Computer Science, 9281 (2015), pp. 469–476. (Citado nas páginas 121, 122 e 140.)
- [68] MATHWORKS®, *Function Reference (R2019a)*, 2019. Disponível em: [https://www.mathworks.com/help/releases/R2019a/pdf\\_doc/matlab/matlab\\_refbook.pdf](https://www.mathworks.com/help/releases/R2019a/pdf_doc/matlab/matlab_refbook.pdf). Acessado em: 16/12/2021. (Citado na página 206.)
- [69] L. MIRSKY, *Symmetric gauge functions and unitarily invariant norms*, The Quarterly Journal of Mathematics, 11 (1960), pp. 50–59. (Citado na página 53.)

- [70] J. J. MORÉ AND S. M. WILD, *Benchmarking derivative-free optimization algorithms*, SIAM Journal on Optimization, 20 (2009), pp. 172–191. (Citado nas páginas 72 e 147.)
- [71] I. X. M. NASCIMENTO, *Otimização sem derivadas: Sobre a construção e a qualidade de modelos quadráticos na solução de problemas irrestritos*, dissertação de mestrado, Universidade Estadual de Campinas, 2014. DOI: 10.47749/T/UNICAMP.2014.929914. Disponível em <https://hdl.handle.net/20.500.12733/1622964>. Acessado em: 16/12/2021. (Citado na página 72.)
- [72] I. X. M. NASCIMENTO, S. A. SANTOS, AND P. J. S. SILVA, *On the sparse and low-rank matrix decomposition problem with examples in video surveillance*, Manuscript in preparation, (2022). (Citado na página 26.)
- [73] Y. NESTEROV, *A method for solving the convex programming problem with convergence rate  $O(1/k^2)$* , Dokl. Akad. Nauk SSSR, 27 (1983), pp. 372–376. (English. Russian original). Translation from Dokl. Akad. Nauk SSSR 269, 543-547 (1983). (Citado na página 22.)
- [74] Y. NESTEROV, *Smooth minimization of non-smooth functions*, Mathematical Programming, 103 (2005), pp. 127–152. (Citado nas páginas 22, 24 e 47.)
- [75] ———, *Gradient methods for minimizing composite functions*, Math. Program., 140 (2013), pp. 125–161. (Citado na página 22.)
- [76] F. NIE, H. HUANG, AND C. DING, *Low-rank matrix recovery via efficient Schatten  $p$ -norm minimization*, in AAAI Conference on Artificial Intelligence, North America, July 2012. (Citado na página 31.)
- [77] N. PARIKH AND S. BOYD, *Proximal algorithms*, Foundations and Trends in Optimization, 1 (2014), pp. 127–239. (Citado nas páginas 56 e 145.)
- [78] ———, *Proximal algorithms (Errata and MATLAB codes)*, 2014. Disponível em: [https://web.stanford.edu/~boyd/papers/prox\\_algs/](https://web.stanford.edu/~boyd/papers/prox_algs/). Acessado em: 16/12/2021. (Citado na página 56.)
- [79] G. B. PASSTY, *Ergodic convergence to a zero of the sum of monotone operators in Hilbert space*, Journal of Mathematical Analysis and Applications, 72 (1979), pp. 383–390. (Citado na página 63.)
- [80] K. PEARSON, *LIII. on lines and planes of closest fit to systems of points in space*, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 2 (1901), pp. 559–572. (Citado na página 23.)

- [81] “*píxel*”, in *Dicionário Priberam da Língua Portuguesa (em linha)*, 2008–2021. <https://dicionario.priberam.org/p%C3%ADxel>. Acessado em: 16/12/2021. (Citado na página 201.)
- [82] M. J. D. POWELL, *A method for nonlinear constraints in minimization problems*, Optimization, (1969), pp. 283–298. (Citado na página 56.)
- [83] M. RAYDAN, *The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem*, SIAM Journal on Optimization, 7 (1997), pp. 26–33. (Citado nas páginas 24 e 59.)
- [84] B. RECHT, M. FAZEL, AND P. A. PARRILO, *Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization*, SIAM Review, 52 (2010), pp. 471–501. (Citado nas páginas 19, 21, 22, 37 e 38.)
- [85] R. T. ROCKAFELLAR, *Conjugates and Legendre transforms of convex functions*, Canadian Journal of Mathematics, 19 (1967), pp. 200–205. (Citado na página 35.)
- [86] R. T. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, Princeton, 1970. (Citado na página 36.)
- [87] D. R. SAHU, J. C. YAO, M. VERMA, AND K. K. SHUKLA, *Convergence rate analysis of proximal gradient methods with applications to composite minimization problems*, Optimization, 70 (2021), pp. 75–100. (Citado na página 21.)
- [88] SBI, *Scene background initialization dataset*. Disponível em: <https://sbmi2015.na.icar.cnr.it/SBIdataset.html>. Acessado em: 16/12/2021. (Citado nas páginas 121, 123, 124 e 140.)
- [89] E. SCHMIDT, *Zur theorie der linearen und nicht linearen integralgleichungen zweite abhandlung*, Mathematische Annalen, 64 (1907), pp. 161–174. (Citado na página 53.)
- [90] M. SCHULZEDARUP, G. BOOK, D. E. QUEVEDO, AND M. NAGAHARA, *Fast hands-off control using ADMM real-time iterations*, IEEE Transactions on Automatic Control, (2021). (Citado na página 25.)
- [91] A. C. SOBRAL, *Robust low-rank and sparse decomposition for moving object detection: from matrices to tensors*, PhD thesis, Université de La Rochelle, 2017. (Citado na página 23.)
- [92] G. STRANG, *Linear Algebra and Learning From Data*, Wellesley-Cambridge Press, Wellesley, Massachusetts, 2019. (Citado na página 53.)

- [93] J. F. STURM, *Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones*, Optimization Methods and Software, 11 (1999), pp. 625–653. (Citado na página 21.)
- [94] K.-C. TOH, M. J. TODD, AND R. H. TÜTÜNCÜ, *On the implementation and usage of SDPT3 – a Matlab software package for semidefinite-quadratic-linear programming, version 4.0*, in Handbook on semidefinite, conic and polynomial optimization, vol. 166, Springer US, Boston, MA, 2012, pp. 715–754. (Citado na página 21.)
- [95] L. VANDENBERGHE AND S. BOYD, *Semidefinite programming*, SIAM Rev., 38 (1996), pp. 49–95. (Citado na página 21.)
- [96] J. VON NEUMANN, *Some matrix-inequalities and metrization of matrix space*, Tomsk University Review, 1 (1937), pp. 286–300. CollectedWorks, Pergamon, Oxford, 1962, Volume IV, 205–218. (Citado na página 38.)
- [97] Y. WANG, W. YIN, AND J. ZENG, *Global convergence of ADMM in nonconvex nonsmooth optimization*, Journal of Scientific Computing, 78 (2019), pp. 29–63. (Citado na página 25.)
- [98] G. A. WATSON, *Characterization of the subdifferential of some matrix norms*, Linear Algebra and its Applications, 170 (1992), pp. 33–45. (Citado nas páginas 37 e 38.)
- [99] G. A. WATSON, *On matrix approximation problems with Ky Fan  $k$  norms*, Numer Algor, 5 (1993), pp. 263–272. (Citado nas páginas 31 e 42.)
- [100] J. WRIGHT, A. GANESH, S. R. RAO, Y. PENG, AND Y. MA, *Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization.*, in Advances in Neural Information Processing Systems 22 (NIPS), vol. 58, 2009, pp. 289–298. (Citado na página 22.)
- [101] S. J. WRIGHT, R. D. NOWAK, AND M. A. T. FIGUEIREDO, *Sparse reconstruction by separable approximation*, IEEE Transactions on Signal Processing, 57 (2009), pp. 2479–2493. (Citado nas páginas 25, 57 e 60.)
- [102] W. YIN, S. OSHER, D. GOLDFARB, AND J. DARBON, *Bregman iterative algorithms for  $\ell_1$ -minimization with applications to compressed sensing*, SIAM J. Imaging Sci., 1 (2008), pp. 143–168. (Citado nas páginas 21 e 22.)
- [103] X. YUAN AND J. YANG, *Sparse and low-rank matrix decomposition via alternating direction methods*, Pacific J. Optim., 9 (2013), pp. 167–180. (Citado nas páginas 11, 12, 22, 25, 56, 57, 68, 74, 77, 78, 79, 80, 93, 131, 139, 154, 155 e 222.)
- [104] K. ZIĘTAK, *On the characterization of the extremal points of the unit sphere of matrices*, Linear Algebra and its Applications, 106 (1988), pp. 57–75. (Citado na página 33.)

## Apêndices

## APÊNDICE A – ALGUMAS CONTAS

$$\underset{X \in \mathbb{R}^{m \times n}}{\text{minimizar}} \left[ \Psi(X; \mu) \stackrel{\text{def}}{=} \Psi(X) = s_\mu(X - A) + c(X) \right], \quad \text{Problema Suavizado.}$$

$$\underset{X}{\text{minimizar}} \left[ Q_k(X) \stackrel{\text{def}}{=} q_k(X) + \omega c(X) \right], \quad \text{Modelo local para o Problema Suavizado.}$$

$$\begin{aligned} q_k(X) &= s_\mu(X^{(k)} - A) + \left\langle \nabla s_\mu(X^{(k)} - A), X - X^{(k)} \right\rangle + \frac{\alpha_k}{2} \|X - X^{(k)}\|_F^2 \\ &= s_\mu(X^{(k)} - A) + \frac{\alpha_k}{2} \left\{ \|X - X^{(k)}\|_F^2 + 2 \left\langle \frac{1}{\alpha_k} \nabla s_\mu(X^{(k)} - A), X - X^{(k)} \right\rangle \right\} \\ &= s_\mu(X^{(k)} - A) + \frac{\alpha_k}{2} \left\{ \|X - X^{(k)}\|_F^2 + 2 \left\langle \frac{1}{\alpha_k} \nabla s_\mu(X^{(k)} - A), X - X^{(k)} \right\rangle + \left\| \frac{1}{\alpha_k} \nabla s_\mu(X^{(k)} - A) \right\|_F^2 \right\} \\ &= s_\mu(X^{(k)} - A) + \frac{\alpha_k}{2} \left\{ \|X - X^{(k)}\|_F^2 + 2 \left\langle \frac{1}{\alpha_k} \nabla s_\mu(X^{(k)} - A), X - X^{(k)} \right\rangle + \left\| \frac{1}{\alpha_k} \nabla s_\mu(X^{(k)} - A) \right\|_F^2 \right\} \\ &= s_\mu(X^{(k)} - A) + \frac{\alpha_k}{2} \left\{ \|X - X^{(k)}\|_F^2 + 2 \left\langle X - X^{(k)}, \frac{1}{\alpha_k} \nabla s_\mu(X^{(k)} - A) \right\rangle + \left\| \frac{1}{\alpha_k} \nabla s_\mu(X^{(k)} - A) \right\|_F^2 \right\} \\ &= s_\mu(X^{(k)} - A) + \frac{\alpha_k}{2} \left\| \left( X - X^{(k)} \right) + \frac{1}{\alpha_k} \nabla s_\mu(X^{(k)} - A) \right\|_F^2 \\ &= s_\mu(X^{(k)} - A) + \frac{\alpha_k}{2} \left\| \left( X - X^{(k)} \right) + \frac{1}{\alpha_k} \nabla s_\mu(X^{(k)} - A) \right\|_F^2 - \frac{1}{2\alpha_k} \left\| \nabla s_\mu(X^{(k)} - A) \right\|_F^2 \\ &= s_\mu(X^{(k)} - A) - \frac{1}{2\alpha_k} \left\| \nabla s_\mu(X^{(k)} - A) \right\|_F^2 + \frac{\alpha_k}{2} \left\| X - X^{(k)} + \frac{1}{\alpha_k} \nabla s_\mu(X^{(k)} - A) \right\|_F^2. \end{aligned}$$

constantes c.r.a.X

$$\begin{aligned}
 & \arg \min_X q_k(X) + c(X) \\
 &= \arg \min_X \overbrace{s_\mu(X^{(k)} - A) - \frac{1}{2\alpha_k} \left\| \nabla_{s_\mu}(X^{(k)} - A) \right\|_F^2 + \frac{\alpha_k}{2} \left\| X - X^{(k)} + \frac{1}{\alpha_k} \nabla_{s_\mu}(X^{(k)} - A) \right\|_F^2}_{\text{constantes c.r.a } X} + c(X) \\
 &= \arg \min_X c(X) + \frac{\alpha_k}{2} \left\| X - \left( X^{(k)} - \frac{1}{\alpha_k} \nabla_{s_\mu}(X^{(k)} - A) \right) \right\|_F^2 \\
 &= \arg \min_X c(X) + \frac{\alpha_k}{2} \left\| X - U^k \right\|_F^2, \quad \text{em que } U^k = X^{(k)} - \frac{1}{\alpha_k} \nabla_{s_\mu}(X^{(k)} - A), \\
 &= \text{prox}_{\frac{c}{\alpha_k}} \left( X^{(k)} - \frac{1}{\alpha_k} \nabla_{s_\mu}(X^{(k)} - A) \right).
 \end{aligned}
 \tag{A.1}$$

$$\tag{A.2}$$

$$\tag{A.3}$$

$$\tag{A.4}$$

# APÊNDICE B – CÓDIGOS DE GERAÇÃO DAS INSTÂNCIAS ALEATÓRIAS NO MATLAB

---

## Algoritmo 5 – Construção de Instância Aleatória de Candès *et al.* no MATLAB

---

```

1 entrada: dimensão  $n$ , número  $s$  de entradas não nulas da componente  $S^*$ , posto
    $r$  da componente  $L^*$ .
2 saída : componentes matriciais  $S^*$  e  $L^*$ , e matriz de dados  $A$ .
3 % inicialização de  $S$  como matriz nula
4  $S = \text{zeros}(m,n)$ ;
5 %  $L$  é o produto de duas matrizes construídas de uma Distribuição Normal
6  $L = \text{normrnd}(0,1/\text{sqrt}(n),n,r)*\text{normrnd}(0,1/\text{sqrt}(n),n,r)$ ;
7 % permutação aleatória das posições matriciais, das quais serão selecionadas  $s$ 
8  $p = \text{randperm}(n^2)$ ;
9 % preenchendo os  $s$  elementos sorteados para serem não nulos
10  $S(p(1:s)) = \text{sign}(\text{randn}(s,1))$ ;
11  $A = S+L$ ;
12 retorne  $S, L$  e  $A$ , e termine.

```

---



---

## Algoritmo 6 – Construção de Instância Impulsiva de Yuan e Yang no MATLAB

---

```

1 entrada: dimensão  $n$ , número  $s$  de entradas não nulas da componente  $S^*$ , posto
    $r$  da componente  $L^*$ .
2 saída : componentes matriciais  $S^*$  e  $L^*$ , e matriz de dados  $A$ .
3 % inicialização de  $S$  como matriz nula
4  $S = \text{zeros}(m,n)$ ;
5 %  $L$  é o produto de duas matrizes construídas da Distribuição Normal Padrão
6  $L = \text{randn}(n,r)*\text{randn}(r,n)$ ;
7 % permutação aleatória das posições matriciais, das quais serão selecionadas  $s$ 
8  $p = \text{randperm}(n^2)$ ;
9 % a magnitude do impulso de  $S$  é a maior entrada de  $L$  em módulo
10  $\text{mag} = \max(\text{abs}(L(:)))$ ;
11 % preenchendo os  $s$  elementos sorteados para serem não nulos
12  $S(p(1:s)) = \text{mag}.*\text{sign}(\text{randn}(s,1))$ ;
13  $A = S+L$ ;
14 retorne  $S, L$  e  $A$ , e termine.

```

---

---

**Algoritmo 7** – Construção de Instância Gaussiana de Yuan e Yang no MATLAB

---

```
1 entrada: dimensão  $n$ , número  $s$  de entradas não nulas da componente  $S^*$ , posto  
    $r$  da componente  $L^*$ .  
2 saída : componentes matriciais  $S^*$  e  $L^*$ , e matriz de dados  $A$ .  
3 % inicialização de  $S$  como matriz nula  
4  $S = \text{zeros}(m,n)$ ;  
5 %  $L$  é o produto de duas matrizes construídas da Distribuição Normal Padrão  
6  $L = \text{randn}(n,r)*\text{randn}(r,n)$ ;  
7 % permutação aleatória das posições matriciais, das quais serão selecionadas  $s$   
8  $p = \text{randperm}(n^2)$ ;  
9 % preenchendo os  $s$  elementos sorteados para serem não nulos  
10  $S(p(1:s)) = \text{randn}(s,1)$ ;  
11  $A = S+L$ ;  
12 retorne  $S, L$  e  $A$ , e termine.
```

---

# APÊNDICE C – DADOS DE ESPARSIDADE E POSTO DAS INSTÂNCIAS ALEATÓRIAS

Tabela 21 – Diferentes medições para a esparsidade das soluções dos problemas aleatórios do grupo CLMW. A coluna “ $n$ ” se refere à dimensão do problema. Nas colunas sob “zeros” constam a quantidade de elementos maiores em módulo do que as tolerâncias indicadas, para cada matriz do problema. Nas colunas sob “spr” estão as taxas de esparsidade, também segundo as tolerâncias indicadas. Os valores das tolerâncias são  $\tau_0^{sp} = 0$  e  $\tau_1^{sp} = 10^{-3}$ .

Instância	$n$	zeros						spr					
		$\tau_0^{sp}$			$\tau_1^{sp}$			$\tau_0^{sp}$			$\tau_1^{sp}$		
		A	S	L	A	S	L	A	S	L	A	S	L
CLMW-1	500	0	237500	0	20279	237500	21284	100	5	100	91.9	5	91.5
CLMW-2	500	0	225000	0	19244	225000	21388	100	10	100	92.3	10	91.4
CLMW-3	1000	0	950000	0	112000	950000	117950	100	5	100	88.8	5	88.2
CLMW-4	1000	0	900000	0	106583	900000	118476	100	10	100	89.3	10	88.2
CLMW-5	2000	0	3800000	0	620210	3800000	652778	100	5	100	84.5	5	83.7
CLMW-6	2000	0	3600000	0	588084	3600000	653822	100	10	100	85.3	10	83.7
CLMW-7	3000	0	8550000	0	1699219	8550000	1788188	100	5	100	81.1	5	80.1
CLMW-8	3000	0	8100000	0	1607892	8100000	1786243	100	10	100	82.1	10	80.2

Correspondência rótulo-instância: veja Tabela 1.

Tabela 22 – Diferentes medições para o posto das soluções dos problemas aleatórios do grupo CLMW. A coluna “ $n$ ” se refere à dimensão do problema. Nas colunas sob “posto” constam a quantidade de valores singulares maiores do que as tolerâncias indicadas, para cada matriz do problema. Nas colunas sob “rkr” estão os postos percentuais, também segundo as tolerâncias indicadas. A primeira tolerância, usada por padrão no MATLAB, é  $\tau_0^{rk} = n \cdot \epsilon$ , em que  $\epsilon$  é a distância, no sistema de pontos flutuantes com precisão dupla, entre as representações de  $\|A\|_2$  e o número que o sucede. A outra tolerância adotada é  $\tau_1^{rk} = 10^{-3} \|A\|_2$ .

Instância	$n$	posto						rkr						
		$\tau_0^{rk}$			$\tau_1^{rk}$			$\tau_0^{rk}$			$\tau_1^{rk}$			
		A	S	L	A	S	L	A	S	L	A	S	L	
CLMW-1	500	500	500	25	499	499	25	100	100	100	5	99.8	99.8	5
CLMW-2	500	500	500	25	499	499	25	100	100	100	5	99.8	99.8	5
CLMW-3	1000	1000	1000	50	999	999	50	100	100	100	5	99.9	99.9	5
CLMW-4	1000	1000	1000	50	998	998	50	100	100	100	5	99.8	99.8	5
CLMW-5	2000	2000	2000	100	1998	1998	100	100	100	100	5	99.9	99.9	5
CLMW-6	2000	2000	2000	100	1997	1998	100	100	100	100	5	99.9	99.9	5
CLMW-7	3000	3000	3000	150	2996	2996	150	100	100	100	5	99.9	99.9	5
CLMW-8	3000	3000	3000	150	2996	2996	150	100	100	100	5	99.9	99.9	5

Correspondência rótulo-instância: veja Tabela 1.

Tabela 23 – Diferentes medições para a esparsidade das soluções dos problemas aleatórios do grupo YImp. A coluna “ $n$ ” se refere à dimensão do problema. Nas colunas sob “zeros” constam a quantidade de elementos maiores em módulo do que as tolerâncias indicadas, para cada matriz do problema. Nas colunas sob “spr” estão as taxas de esparsidade, também segundo as tolerâncias indicadas. Os valores das tolerâncias são  $\tau_0^{sp} = 0$  e  $\tau_1^{sp} = 10^{-3}$ .

Instância	$n$	zeros						spr					
		$\tau_0^{sp}$			$\tau_1^{sp}$			$\tau_0^{sp}$			$\tau_1^{sp}$		
		A	S	L	A	S	L	A	S	L	A	S	L
YImp-1	100	0	9500	0	87	9500	92	100	5	100	99.1	5	99.1
YImp-2	100	0	9500	0	60	9500	65	100	5	100	99.4	5	99.4
YImp-3	100	0	9000	0	85	9000	100	100	10	100	99.2	10	99
YImp-4	100	0	9000	0	78	9000	85	100	10	100	99.2	10	99.2
YImp-5	500	0	237500	0	2182	237500	2299	100	5	100	99.1	5	99.1
YImp-6	500	0	237500	0	1801	237500	1899	100	5	100	99.3	5	99.2
YImp-7	500	0	225000	0	1878	225000	2087	100	10	100	99.2	10	99.2
YImp-8	500	0	225000	0	1663	225000	1842	100	10	100	99.3	10	99.3
YImp-9	1000	0	950000	0	7226	950000	7595	100	5	100	99.3	5	99.2
YImp-10	1000	0	950000	0	7150	950000	7516	100	5	100	99.3	5	99.2
YImp-11	1000	0	900000	0	7056	900000	7836	100	10	100	99.3	10	99.2
YImp-12	1000	1	900000	0	6832	900000	7582	100	10	100	99.3	10	99.2

Correspondência rótulo-instância: veja Tabela 1.

Tabela 24 – Diferentes medições para o posto das soluções dos problemas aleatórios do grupo YImp. A coluna “ $n$ ” se refere à dimensão do problema. Nas colunas sob “posto” constam a quantidade de valores singulares maiores do que as tolerâncias indicadas, para cada matriz do problema. Nas colunas sob “rkr” estão os postos percentuais, também segundo as tolerâncias indicadas. A primeira tolerância, usada por padrão no MATLAB, é  $\tau_0^{rk} = n \cdot \epsilon$ , em que  $\epsilon$  é a distância, no sistema de pontos flutuantes com precisão dupla, entre as representações de  $\|A\|_2$  e o número que o sucede. A outra tolerância adotada é  $\tau_1^{rk} = 10^{-3} \|A\|_2$ .

Instância	$n$	posto						rkr					
		$\tau_0^{rk}$		$\tau_1^{rk}$		$\tau_0^{rk}$		$\tau_1^{rk}$		$\tau_0^{rk}$		$\tau_1^{rk}$	
		A	S	L	A	S	L	A	S	L	A	S	L
YImp-1	100	100	100	5	99	100	5	100	100	5	99	100	5
YImp-2	100	100	10	10	100	100	10	100	100	10	100	100	10
YImp-3	100	100	100	5	100	99	5	100	100	5	100	99	5
YImp-4	100	100	100	10	100	100	10	100	100	10	100	100	10
YImp-5	500	500	500	25	498	498	25	100	100	5	99.6	99.6	5
YImp-6	500	500	500	50	499	499	50	100	100	10	99.8	99.8	10
YImp-7	500	500	500	25	499	498	25	100	100	5	99.8	99.6	5
YImp-8	500	500	500	50	499	498	50	100	100	10	99.8	99.6	10
YImp-9	1000	1000	1000	50	997	997	50	100	100	5	99.7	99.7	5
YImp-10	1000	1000	1000	100	997	996	100	100	100	10	99.7	99.6	10
YImp-11	1000	1000	1000	50	999	998	50	100	100	5	99.9	99.8	5
YImp-12	1000	1000	1000	100	998	998	100	100	100	10	99.8	99.8	10

Correspondência rótulo-instância: veja Tabela 1.

Tabela 25 – Diferentes medições para a esparsidade das soluções dos problemas aleatórios do grupo YGaus. A coluna “ $n$ ” se refere à dimensão do problema. Nas colunas sob “zeros” constam a quantidade de elementos maiores em módulo do que as tolerâncias indicadas, para cada matriz do problema. Nas colunas sob “spr” estão as taxas de esparsidade, também segundo as tolerâncias indicadas. Os valores das tolerâncias são  $\tau_0^{sp} = 0$  e  $\tau_1^{sp} = 10^{-3}$ .

Instância	$n$	zeros						spr					
		$\tau_0^{sp}$			$\tau_1^{sp}$			$\tau_0^{sp}$			$\tau_1^{sp}$		
		A	S	L	A	S	L	A	S	L	A	S	L
YGaus-1	100	0	9500	0	73	9506	71	100	5	100	99.3	4.94	99.3
YGaus-2	100	0	9500	0	39	9507	39	100	5	100	99.6	4.93	99.6
YGaus-3	100	0	9000	0	47	9008	52	100	10	100	99.5	9.92	99.5
YGaus-4	100	0	9000	0	45	9014	49	100	10	100	99.6	9.86	99.5
YGaus-5	500	0	237500	0	1103	237786	1101	100	5	100	99.6	4.89	99.6
YGaus-6	500	0	237500	0	1052	237875	1068	100	5	100	99.6	4.85	99.6
YGaus-7	500	0	225000	0	1059	225529	1086	100	10	100	99.6	9.79	99.6
YGaus-8	500	0	225000	0	953	225662	962	100	10	100	99.6	9.74	99.6
YGaus-9	1000	0	950000	0	4252	951422	4228	100	5	100	99.6	4.86	99.6
YGaus-10	1000	0	950000	0	4462	952266	4479	100	5	100	99.6	4.77	99.6
YGaus-11	1000	0	900000	0	4626	903227	4678	100	10	100	99.5	9.68	99.5
YGaus-12	1000	0	900000	0	4346	904152	4264	100	10	100	99.6	9.58	99.6

Correspondência rótulo-instância: veja Tabela 1.

Tabela 26 – Diferentes medições para o posto das soluções dos problemas aleatórios do grupo  $YY_{\text{gaus}}$ . A coluna “ $n$ ” se refere à dimensão do problema. Nas colunas sob “posto” constam aq quantidade de valores singulares maiores do que as tolerâncias indicadas, para cada matriz do problema. Nas colunas sob “rkr” estão os postos percentuais, também segundo as tolerâncias indicadas. A primeira tolerância, usada por padrão no MATLAB, é  $\tau_0^{rk} = n \cdot \epsilon$ , em que  $\epsilon$  é a distância, no sistema de pontos flutuantes com precisão dupla, entre as representações de  $\|A\|_2$  e o número que o sucede. A outra tolerância adotada é  $\tau_1^{rk} = 10^{-3} \|A\|_2$ .

Instância	$n$	posto						rkr					
		$\tau_0^{rk}$			$\tau_1^{rk}$			$\tau_0^{rk}$			$\tau_1^{rk}$		
		A	S	L	A	S	L	A	S	L	A	S	L
$YY_{\text{gaus-1}}$	100	100	100	5	94	93	5	100	100	5	94	93	5
$YY_{\text{gaus-2}}$	100	100	10	10	92	90	10	100	100	10	92	90	10
$YY_{\text{gaus-3}}$	100	100	100	5	97	96	5	100	100	5	97	96	5
$YY_{\text{gaus-4}}$	100	100	100	10	96	95	10	100	100	10	96	95	10
$YY_{\text{gaus-5}}$	500	500	25	25	459	457	25	100	100	5	91.8	91.4	5
$YY_{\text{gaus-6}}$	500	500	50	50	453	449	50	100	100	10	90.6	89.8	10
$YY_{\text{gaus-7}}$	500	500	25	25	469	469	25	100	100	5	93.8	93.8	5
$YY_{\text{gaus-8}}$	500	500	50	50	468	466	50	100	100	10	93.6	93.2	10
$YY_{\text{gaus-9}}$	1000	1000	50	50	882	879	50	100	100	5	88.2	87.9	5
$YY_{\text{gaus-10}}$	1000	1000	100	100	871	864	100	100	100	10	87.1	86.4	10
$YY_{\text{gaus-11}}$	1000	1000	50	50	917	913	50	100	100	5	91.7	91.3	5
$YY_{\text{gaus-12}}$	1000	1000	100	100	914	908	100	100	100	10	91.4	90.8	10

Correspondência rótulo-instância: veja Tabela 1.

# APÊNDICE D – DADOS DE ESPARSIDADE E POSTO DAS INSTÂNCIAS DOS VÍDEOS

Tabela 27 – Dados de esparsidade da matriz de entrada  $A = (a_{ij}) \in [0, 1]^{m \times n}$  e da respectiva Solução de Referência ( $S, L$ ) das instâncias do Grupo CAVIAR no Espaço de Otimização: “zeros” é a quantidade de elementos numericamente nulos (com magnitude inferior ou igual a uma das duas tolerância indicadas) e “spr” é a esparsidade percentual correspondente, isto é, a porcentagem de elementos considerados não nulos. As tolerâncias adotadas foram  $\tau_0^{sp} = 0$  e  $\tau_1^{sp} = 10^{-3} \cdot \max\{|a_{ij}|\}$ .

Instância	m	n	zeros						taxa de esparsidade (spr)							
			$\tau_0^{sp}$			$\tau_1^{sp}$			$\tau_0^{sp}$			$\tau_1^{sp}$				
			A	S	L	A	S	L	A	S	L	A	S	L		
v11	110592	383	155856	7662	7660	155856	21149978	134249	99.6	100	100	99.6	100	100	50.1	99.7
v12	110592	750	457730	38259	38250	457730	41428649	398354	99.4	100	100	99.4	100	100	50.1	99.5
v13	110592	450	179565	5403	5400	179565	24756402	155276	99.6	100	100	99.6	100	100	50.3	99.7
v14	110592	750	305550	14257	14250	305550	42711648	265718	99.6	100	100	99.6	100	100	48.5	99.7
v21	110592	750	426391	12007	12000	426391	37955529	379723	99.5	100	100	99.5	100	100	54.2	99.5
v22	110592	750	410074	15754	15750	410074	38101879	353942	99.5	100	100	99.5	100	100	54.1	99.6
v23	110592	560	319381	11766	11760	319381	28826438	278391	99.5	100	100	99.5	100	100	53.5	99.6
v24	110592	750	426949	12757	12750	426949	37548742	366031	99.5	100	100	99.5	100	100	54.7	99.6
v25	110592	750	433791	13502	13500	433791	38004420	380388	99.5	100	100	99.5	100	100	54.2	99.5
v31	110592	492	14068	1	0	14068	17674267	1886	100	100	100	100	100	100	67.5	100
v32	110592	750	657160	6	0	657160	27529251	666263	99.2	100	100	99.2	100	100	66.8	99.2
v33	110592	750	465050	2	0	465050	26543536	455069	99.4	100	100	99.4	100	100	68	99.5
v34	110592	606	654042	4	0	654042	21765152	667422	99	100	100	99	100	100	67.5	99
v35	110592	959	645	4	0	645	35783483	0	100	100	100	100	100	100	66.3	100

Tabela 28 – Dados de posto da matriz de entrada  $A = (a_{ij}) \in [0, 1]^{m \times n}$  e da respectiva Solução de Referência ( $S, L$ ) das instâncias do Grupo CAVIAR no Espaço de Otimização: “posto” é a quantidade de valores singulares com valor superior à tolerância indicada) e “rkr” é o posto percentual correspondente, isto é, o posto dividido pela menor dimensão do problema, em porcentagem. As tolerâncias adotadas foram  $\tau_0^{sp} = \max\{m, n\} \cdot \epsilon(\sigma_1(A))$  e  $\tau_1^{sp} = 10^{-3} \sigma_1(A)$ , em que  $\sigma_1(A)$  é o maior valor singular de  $A$  e  $\epsilon(x)$  é a distância entre  $x$  e seu consecutivo na lista crescente de números exatamente representáveis em precisão dupla no sistema de ponto flutuante.

Instância	m	n	posto						posto percentual (rkr)					
			$\tau_0^{r,k}$			$\tau_1^{r,k}$			$\tau_0^{r,k}$			$\tau_1^{r,k}$		
			A	S	L	A	S	L	A	S	L	A	S	L
v11	110592	383	380	184	115	106	23	99.2	99.2	48	30	27.7	6.01	
v12	110592	750	744	372	259	249	48	99.2	99.2	49.6	34.5	33.2	6.4	
v13	110592	450	449	219	135	126	27	99.8	99.8	48.7	30	28	6	
v14	110592	750	744	369	247	237	44	99.2	99.2	49.2	32.9	31.6	5.87	
v21	110592	750	737	370	321	316	74	98.3	98.3	49.3	42.8	42.1	9.87	
v22	110592	750	744	378	222	213	48	99.2	99.2	50.4	29.6	28.4	6.4	
v23	110592	560	558	282	187	180	35	99.6	99.6	50.4	33.4	32.1	6.25	
v24	110592	750	745	382	186	178	43	99.3	99.3	50.9	24.8	23.7	5.73	
v25	110592	750	743	381	208	200	45	99.1	99.1	50.8	27.7	26.7	6	
v31	110592	492	492	233	244	213	37	100	100	47.4	49.6	43.3	7.52	
v32	110592	750	749	370	282	257	22	99.9	99.9	49.3	37.6	34.3	2.93	
v33	110592	750	749	352	267	243	24	99.9	99.9	46.9	35.6	32.4	3.2	
v34	110592	606	605	285	264	242	31	99.8	99.8	47	43.6	39.9	5.12	
v35	110592	959	959	517	377	341	43	100	100	53.9	39.3	35.6	4.48	

Tabela 29 – Dados de esparsidade da matriz de entrada  $A = (a_{ij}) \in [0, 1]^{m \times n}$  e da respectiva Solução de Referência  $(S, L)$  das instâncias do Grupo SBI no Espaço de Otimização: “zeros” é a quantidade de elementos numericamente nulos (com magnitude inferior ou igual a uma das duas tolerância indicadas) e “spr” é a esparsidade percentual correspondente, isto é, a porcentagem de elementos considerados não nulos. As tolerâncias adotadas foram  $\tau_0^{sp} = 0$  e  $\tau_1^{sp} = 10^{-3} \cdot \max\{|a_{ij}|\}$ .

Instância	m	n	zeros						taxa de esparsidade (spr)						
			$\tau_0^{sp}$			$\tau_1^{sp}$			$\tau_0^{spr}$			$\tau_1^{spr}$			
			A	S	L	A	S	L	A	S	L	A	S	L	
Board	32800	228	5	1	0	5	2720678	0	100	100	100	100	100	63.6	100
Candela_m1.10	101376	350	238	1	0	238	20033180	36	100	100	100	100	100	43.5	100
CAVIAR1	98304	610	0	0	0	0	26716696	0	100	100	100	100	100	55.4	100
CAVIAR2	98304	460	0	0	0	0	21511496	0	100	100	100	100	100	52.4	100
CaVignal	27200	258	39	0	0	39	5683483	0	100	100	100	100	100	19	100
Foliage	28800	394	131	1	0	131	2897719	27	100	100	100	100	100	74.5	100
HallAndMonitor	84480	296	29910	1	0	29910	5091043	5837	99.9	100	100	100	100	79.6	100
HighwayI	76800	440	37	1	0	37	7244929	0	100	100	100	100	100	78.6	100
HighwayII	76800	500	297	0	0	297	5445856	0	100	100	100	100	100	85.8	100
HumanBody2	76800	740	2	3	0	2	18131835	0	100	100	100	100	100	68.1	100
IBMtest2	76800	90	39845	1	0	39845	1023226	19312	99.4	100	100	100	100	85.2	99.7
PeopleAndFoliage	76800	341	1470	0	0	1470	7668329	222	100	100	100	100	100	70.7	100
Snellen	20736	321	8390	1	0	8390	1890021	2484	99.9	100	100	100	100	71.6	100
Toscana	480000	6	6	0	0	6	682615	0	100	100	100	100	100	76.3	100

Tabela 30 – Dados de posto da matriz de entrada  $A = (a_{ij}) \in [0, 1]^{m \times n}$  e da respectiva Solução de Referência ( $S, L$ ) das instâncias do Grupo SBI no Espaço de Otimização: “posto” é a quantidade de valores singulares com valor superior à tolerância indicada) e “rkr” é o posto percentual correspondente, isto é, o posto dividido pela menor dimensão do problema, em porcentagem. As tolerâncias adotadas foram  $\tau_0^{sp} = \max\{m, n\} \cdot \epsilon(\sigma_1(A))$  e  $\tau_1^{sp} = 10^{-3} \sigma_1(A)$ , em que  $\sigma_1(A)$  é o maior valor singular de  $A$  e  $\epsilon(x)$  é a distância entre  $x$  e seu consecutivo na lista crescente de números exatamente representáveis em precisão dupla no sistema de ponto flutuante.

Instância	m	n	posto						posto percentual (rkr)					
			$\tau_0^{rk}$			$\tau_1^{rk}$			$\tau_0^{rk}$			$\tau_1^{rk}$		
			A	S	L	A	S	L	A	S	L	A	S	L
Board	32800	228	228	119	228	228	228	59	100	100	52.2	100	100	25.9
Candela_m1.10	101376	350	350	196	188	181	18	100	100	56	53.7	51.7	51.4	
CAVIAR1	98304	610	600	296	209	202	38	98.4	98.4	48.5	34.3	33.1	6.23	
CAVIAR2	98304	460	460	219	171	163	29	100	100	47.6	37.2	35.4	6.3	
CaVignal	27200	258	258	72	113	111	13	100	100	27.9	43.8	43	5.04	
Foliage	28800	394	394	210	391	390	137	100	100	53.3	99.2	99	34.8	
HallAndMonitor	84480	296	296	118	296	296	13	100	100	39.9	100	100	4.39	
HighwayI	76800	440	440	217	440	440	67	100	100	49.3	100	100	15.2	
HighwayII	76800	500	500	289	500	500	8	100	100	57.8	100	100	1.6	
HumanBody2	76800	740	740	353	670	669	82	100	100	47.7	90.5	90.4	11.1	
IBMtest2	76800	90	90	25	90	90	10	100	100	27.8	100	100	11.1	
PeopleAndFoliage	76800	341	341	175	341	341	82	100	100	51.3	100	100	24	
Snellen	20736	321	321	174	320	320	107	100	100	54.2	99.7	99.7	33.3	
Toscana	480000	6	6	3	6	6	3	100	100	50	100	100	50	

# APÊNDICE E – IMAGENS E VÍDEOS DIGITAIS

Os processos envolvidos na aquisição e no processamento digital de imagens são complexos e diversificados. O que apresentamos a continuação são noções gerais sobre alguns aspectos desses processos, suficientemente simplificadas para o nosso contexto. Caso o leitor queira se aprofundar no tema, indicamos o Capítulo 2 de Gonzalez e Woods [42], em particular, a Seção 2.3.

A maneira mais comum de lidar computacionalmente com uma imagem bidimensional, que a princípio é um objeto visual de caráter plano e contínuo, é por meio da amostragem de um conjunto finito de pontos que a represente. Frequentemente, o que se faz é adotar uma malha retangular uniforme, por exemplo, dividindo a imagem em linhas horizontais e verticais que formam retângulos cujos centros (ou vértices) são os pontos do conjunto amostral, formando assim uma estrutura de matriz de pontos que em Computação Gráfica se chama *gráfico de rastreo* (*raster image* ou, ainda, *bitmap*, ambos em inglês). Cada um dos pontos utilizados para representar o gráfico de rastreo é chamado de *píxel*, forma aportuguesada da palavra inglesa *pixel*, que por sua vez vem da expressão *picture element*, “elemento de imagem”, em tradução direta. O plural da palavra aportuguesada *píxel* é, portanto, *píxeis* (cf. [81]). Chamamos de *resolução* de uma imagem a dimensão da matriz de *píxeis* que a representa ou, por extensão, a quantidade de *píxeis* dessa matriz. De um modo geral, resoluções mais altas indicam uma melhor capacidade da imagem digital em se assemelhar visualmente à imagem real, embora mais custoso seja processá-la e armazená-la.

Cada ponto assume uma única cor, sendo que a quantidade de cores possíveis depende da quantidade de informação (e, conseqüentemente, de memória) que se reserva para isso. Por ser corriqueiro em sistemas de videovigilância, delimitamos o escopo deste trabalho a imagens de 8 *bits* monocromáticas (coloridas apenas com tons de cinza). Isso significa que a cada um dos pontos amostrais atribuímos um valor numérico inteiro dentre as  $2^8 = 256$  intensidades disponíveis para a cor cinza, iniciando em zero (0) com a cor preta e chegando a 255 com a cor branca. Geralmente, a representação das intensidades de cor é reescalada do intervalo [0, 255] para o intervalo [0, 1] mediante um escalamento uniforme (transformação linear), sem alteração na quantidade de cores representáveis, com o zero (0) se mantendo como preto e o 1 se tornando o branco. Com os valores de intensidades de cor para cada ponto é possível construir uma matriz numérica cujos elementos têm posições que de alguma

forma correspondem às coordenadas espaciais dos pontos na imagem. A essa matriz numérica dá-se o nome de *imagem digital*.

No caso de imagens multicoloridas, a principal forma de representação digital é o sistema RGB, o qual é fundamentado na Teoria Tricromática da Visão. Segundo essa teoria, as cores que o olho humano padrão<sup>1</sup> é capaz de enxergar são combinações das chamadas cores primárias de luz: vermelho, verde e azul – *red*, *green* e *blue*, em inglês, origem da sigla RGB (cf. [18, Seção VI.1], [42, Seção 7.2]). O padrão RGB é classificado como um *sistema aditivo de cores* pois pressupõe a exibição em um fundo-base preto ao qual são *adicionadas* simultaneamente intensidades das três cores que o compõem. Novamente, a quantidade de cores de exibição depende da quantidade de *bits* destinada para cada píxel. Mantendo os 8 *bits* por cor, tendo agora três cores-base, temos 24 *bits* de informação por píxel e, portanto, um total de  $2^{24} = 16777216$  cores disponíveis. Analogamente às imagens em tons de cinza, imagens coloridas podem ser representadas no computador como três matrizes numéricas – uma matriz-componente para cada cor – também mantendo a correspondência entre as posições dos elementos matriciais com a localização dos píxeis na imagem. Em outras palavras, podemos dizer que a cada píxel associamos uma tripla ordenada que indica a intensidade de cada uma das cores do padrão RGB. É comum a nomenclatura que denota as matrizes das intensidades de vermelho, verde e azul, ou seus elementos em determinado píxel, por R, G e B, respectivamente.

A respeito do termo píxel, vale um adendo: por extensão, também podemos chamar de píxeis os elementos da matriz numérica que representa a imagem digital, no caso da escala de cinza, ou à tripla de valores do padrão RGB, no caso de imagens multicoloridas, sem prejuízos de compreensão. Em outros contextos, píxel pode se referir também à unidade de captação ou emissão de luz em dispositivos de captura e exibição de imagens.

Por sua vez, como já adiantamos, o conceito de vídeo é concebido neste trabalho como uma mera sequência de imagens que, reproduzidas sucessivamente a uma certa frequência, causam a impressão de movimento contínuo. As imagens que compõem um vídeo digital são também chamadas de quadros ou *frames* e a taxa padrão com que são capturadas ou exibidas é medida em *frames* por segundo (fps). O termo *frame*, em inglês, significa fotograma e é utilizado desde a época dos filmes analógicos, sendo bastante frequente o uso da palavra em português como anglicismo. Assim como com as imagens digitais, nosso interesse está nos vídeos em escala de cinza.

---

<sup>1</sup> Sempre que abordarmos características e capacidades do olho humano, por simplicidade nos referimos a um padrão fisiológico idealizado que, incontestavelmente, não contempla toda a população e que depende de incontáveis fatores.

## Indexação de Elementos de uma Imagem Digital

A seguir, usando os conceitos apresentados em [42, Seção 2.4] e [43, Seção 1.7.5], desenvolvemos com mais detalhes a noção de imagem digital e a indexação de seus elementos.

Podemos elaborar o conceito de imagem de diferentes maneiras, abstratas ou concretas. Sem perda de generalidade, uma imagem pode ser idealizada como um objeto bidimensional retangular de caráter contínuo no que se refere ao espaço que ocupa quando colocada no plano. Em se tratando de imagens em escala de cinza, como é de nosso interesse, podemos pensar matematicamente uma imagem como uma função  $p : [0, a] \times [0, b] \rightarrow [0, 1]$  que a cada posição  $(x, y)$  de seu domínio associa uma intensidade de cor  $p(x, y)$  qualquer no intervalo  $[0, 1]$ . Ainda que a intensidade assuma valores em outros intervalos, sempre podemos convertê-la linearmente a esse intervalo unitário. Naturalmente, uma forma de representar a imagem é, portanto, por intermédio de seu gráfico tridimensional. Por convenção e conveniência com a notação matricial aqui adotada, costuma-se representar o domínio de  $p$  num sistema de coordenadas que é resultado de uma rotação do plano cartesiano padrão em  $90^\circ$  no sentido horário. Em outras palavras, o eixo  $x$  passa a ser o eixo vertical, crescendo para baixo, e o eixo  $y$ , o eixo horizontal, crescendo para a direita. Dessa forma, o canto superior esquerdo da imagem é associado à origem do sistema de coordenadas.

Uma conversão em imagem digital pode ocorrer por meio da amostragem da função  $p$  numa malha uniforme de pontos com  $m$  linhas e  $n$  colunas, totalizando  $mn$  elementos de imagem (píxeis). Em termos de memória de armazenamento, o número  $b$  de *bits* necessários para guardar uma imagem digital de dimensão  $m \times n$  na qual cada píxel é representado por  $d$  *bits* - e ou seja, com  $2^d$  tons de cor - é dado pelo produto

$$b = m \cdot n \cdot d.$$

Podemos denotar os pontos da malha por  $(x_i, y_j)$ , em que  $i \in \{1, 2, \dots, m\}$ ,  $j \in \{1, 2, \dots, n\}$  e os valores  $x_i$  e  $y_j$  são ordenados em concordância com os respectivos índices. Note que, de acordo com o sistema de coordenadas que fixamos, variações em  $x_i$  implicam em deslocamentos verticais e variações em  $y_j$  em deslocamentos horizontais; além disso, o ponto  $(x_1, y_1)$  corresponde à origem do sistema de coordenadas, o que não é comum para muitos livros sobre imagens digitais mas é coerente com a notação do MATLAB.

Estruturamos, então, os valores das intensidades de  $p$  nos pontos amostrais como uma matriz  $P = (p_{ij})$  de dimensão  $m \times n$  de tal forma que, sob a conveniência da notação adotada,  $p_{ij} = p(x_i, y_j)$  para todo ponto amostral  $(x_i, y_j)$ . A matriz  $P$  de intensidade de cor também pode ser referida simplesmente como “imagem  $P$ ”, sem prejuízo de entendimento, assim como cada elemento  $p_{ij}$  pode ser chamado de píxel.

Além da indexação de elementos por índice duplo,  $p_{ij}$ , uma outra maneira de referir-se a cada elemento da matriz  $P$  é por meio de um índice único (indexação linear), a qual será indispensável no entendimento da conversão entre a aplicação prática e problema de otimização. A indexação linear se utiliza de apenas um único número para se referir às posições das entradas de uma matriz e é baseada na ordem do elemento quando percorremos completamente cada coluna de cima para baixo, começando da posição  $(1, 1)$ , e prosseguimos para a coluna da direita, até completar a matriz. Há, evidentemente, outras maneiras de se percorrer a matriz, mas essa é a mais convencional para nossos propósitos. Assim, obtemos o índice linear  $\ell$  do elemento de posição  $(i, j)$  em uma matriz  $m \times n$  como segue:

$$\ell = \ell(i, j) = i + m(j - 1). \quad (\text{E.1})$$

Por outro lado, obtemos a posição  $(i, j)$  do elemento de índice linear  $\ell$  nessa mesma matriz fazendo:

$$\begin{aligned} i &= 1 + \text{mod}(\ell - 1, m), \\ j &= 1 + [\ell - \text{mod}(\ell, m)]/m, \end{aligned} \quad (\text{E.2})$$

em que  $\text{mod}(\ell, m)$  é o resto da divisão inteira de  $\ell$  por  $m$ . Na implementação vetorizada do MATLAB, como os índices propriamente ditos dificilmente são necessários, as conversões de um formato para outro são feitas por meio da função nativa `reshape` e suas variações.

Como veremos, a indexação linear de imagens e a indexação dupla de vídeos terá um papel fundamental no rearranjo dos elementos de cada matriz que compõe os vídeos, já que converteremos a matriz em vetor coluna seguindo essa ordenação.

### Indexação de Elementos de um Vídeo Digital

Podemos tratar da indexação de um vídeo digital monocromático, enquanto uma sequência de imagens do mesmo tipo, por meio de matrizes ou *arrays* tridimensionais em que a terceira dimensão indica a ordem de exibição de cada *frame* do vídeo.

Para definir um vídeo como função, basta alterar o domínio da função  $p$  introduzida anteriormente de forma a contemplar uma imagem a cada instante de tempo:  $p : [0, a] \times [0, b] \times [0, c] \rightarrow [0, 1]$  que a cada posição  $(x, y)$  de seu domínio espacial, num dado instante de tempo  $t$ , associa uma intensidade de cor  $p(x, y, t)$  qualquer no intervalo  $[0, 1]$ .

Procedendo analogamente, agora acrescentando uma discretização temporal  $\{t_1, t_2, \dots, t_{n_f}\}$  à malha tridimensional, temos a relação entre o vídeo  $p$  e vídeo digital

$P = (p_{ijk})$  dada por:

$$p_{ijk} = p(x_i, y_j, t_k), \quad i \in \{1, 2, \dots, m\}, j \in \{1, 2, \dots, n\}, k \in \{1, 2, \dots, n_f\}. \quad (\text{E.3})$$

Portanto, o vídeo digital composto por  $n_f$  imagens de dimensão  $m \times n$  e de  $d$  bits por píxel ocupa na memória de um computador o total  $b$  de bits dado por:

$$b = m \cdot n \cdot n_f \cdot d.$$

Assim como foi viável representar os elementos de uma imagem em escala de cinza por meio da indexação linear, podemos também exprimir a posição dos píxeis de um vídeo digital de mesmo tipo por meio de uma indexação dupla, ou seja, com dois índices ( $ij$ ) em vez de três ( $ijk$ ). De fato, isso nos permite também trabalhar com vídeos como se fossem matrizes bidimensionais, o que será fundamental para enquadrar a aplicação prática no problema de otimização.

Nossa estratégia consiste em reorganizar os *frames* do vídeo como se fossem colunas de uma matriz, utilizando para isso a indexação linear de cada *frame*. Portanto, obtemos o índice duplo ( $i, j$ ) para o píxel  $p_{ijk}$  de posição ( $i, j, k$ ) em  $P$  realizando a seguinte conversão:

$$j = k, \quad (\text{E.4})$$

$$i = \ell(i, j), \quad (\text{E.5})$$

em que  $\ell(i, j)$  é a indexação linear apresentada em (E.1). Com o auxílio de (E.2), a conversão inversa é imediata.

Com efeito, poderíamos ir além e pensar em uma indexação linear para vídeos digitais, mas neste trabalho isso não se faz necessário.

### Conversão de RGB para Escala de Cinza

Vários dos vídeos de teste que adotamos em nossos experimentos computacionais foram gravados e disponibilizados no padrão de cores RGB. Assim, cada imagem que forma o vídeo é representada por três matrizes de intensidades,  $R$ ,  $G$  e  $B$ , uma para cada cor do padrão: vermelho, verde e azul, respectivamente. No caso das imagens de 24 bits, essas matrizes possuem como entradas os números inteiros de 0 a 255, já que são reservados 8 bits para cada cor.

Para converter os vídeos de teste à escala de cinza, precisamos realizar transformações nas três matrizes do padrão de forma a combiná-las em uma única matriz de intensidade, então representando os tons de cinza. Há diferentes maneiras de realizar essa conversão, sendo a mais simples aquela que adota para cada entrada da nova matriz a média das três intensidades de cor do píxel correspondente. No

entanto, o olho humano não é igualmente sensível às luzes vermelha, verde e azul, de forma que outras ponderações diferentes para cada cor podem conduzir a melhores resultados visuais [42, Seção 6.1]. A combinação implementada pelo MATLAB em sua função nativa `rgb2gray` [68, p. 1-12987] pondera as três intensidades em uma usando os mesmos coeficientes do sistema NTSC das televisões analógicas, que, apesar de lidar com cores, precisava funcionar também em aparelhos monocromáticos [43, Seção 6.2.1]. A mescla de  $R$ ,  $G$  e  $B$  em uma única matriz  $W$  é feita mediante a seguinte combinação linear convexa, cujos coeficientes são mostrados com três casas decimais de precisão:

$$\widehat{W} = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B. \quad (\text{E.6})$$

Os coeficientes em (E.6) estão relacionados com o cálculo da luminância, uma medida da densidade superficial de intensidade de luz que atravessa ou é emitida de determinada superfície em uma dada direção, de acordo com determinado padrão bastante comum [68, p. 1-12903]. De certa forma, o cálculo da luminância também está relacionado com a quantidade de energia luminosa incidente em um olho humano típico que gera certo conforto visual em situações padrão. O olho humano é mais sensível à cor verde, por isso o valor do coeficiente que multiplica  $G$  é maior, sendo menos sensível às cores vermelha e azul, nessa ordem, o que também corrobora os demais valores. O que a `rgb2gray` faz é eliminar o matiz e a saturação, mantendo apenas a luminância da imagem RGB.

Após a obtenção de uma imagem em escala de cinza, o que fazemos é normalizar seus valores para o intervalo  $[0, 1]$ ,

$$W = \frac{1}{255} \cdot \widehat{W}. \quad (\text{E.7})$$

A nova matriz ( $W$ ) é armazenada como um *array* bidimensional de números que são representados por meio do formato de ponto flutuante de precisão dupla (`double`), que é o padrão mais utilizado para representar números reais computacionalmente e realizar cálculos matemáticos.

## Exibição de Imagens no MATLAB

### Imagens no MATLAB

Apresentamos a seguir algumas noções sobre o processamento de imagens no MATLAB que são úteis para o entendimento de nossos experimentos computacionais. Para um tratamento mais aprofundado, indicamos os Capítulos 1 e 6 de [43].

Conforme discutimos, uma imagem digital pode ser classificada de acordo com vários aspectos. Um deles é a resolução de intensidade, ou seja, a quantidade

de *bits* reservada para cada píxel, que usualmente varia entre 8, 16 ou 32 *bits*; o tipo mais comum dentre as imagens, inclusive em registros de videovigilância, onde está nosso interesse, é o monocromático de 8 *bits*, em que há  $2^8 = 256$  diferentes tons de determinada cor, tipicamente cinza. De acordo com a subseção *Spatial and Intensity Resolution* de [42, Seção 2.4], em alguns contextos que requerem detalhamento aprimorado nas variações das cores, é comum o uso de imagens de 16 *bits*, sendo as de 32 *bits* consideravelmente mais raras que as demais. Imagens coloridas segundo o padrão RGB são vistas como três canais (componentes), geralmente de 8 *bits* cada, totalizando 24 *bits* por píxel.

Há também uma outra característica das imagens que está relacionada diretamente com o *software* em que o arquivo é lido e processado: o tipo de dado<sup>1</sup> por meio do qual a imagem é instanciada e operacionalizada. No caso do MATLAB, os dois principais tipos que utilizamos neste trabalho são a `uint8` e a `double`. O tipo `uint8` (*unsigned integer of 8 bits*) armazena números inteiros não negativos no intervalo fechado de 0 a 255, com 8 *bits* por elemento. Já o tipo `double`, de precisão dupla em ponto flutuante, admite números no intervalo de  $-10^{308}$  a  $10^{308}$ , também reservando 64 *bits* para cada elemento. Como podemos intuir, o tipo `double` é muito mais conveniente quando se quer realizar operações matemáticas com os valores, ainda que isso possa ocasionar um aumento no gasto com memória de armazenamento e tempo de processamento. Apesar da enorme faixa de valores representáveis com o tipo `double`, eles não estão igualmente distribuídos ao longo da reta real e, devido a especificidades do sistema de pontos flutuantes, a maior densidade de números não negativos representáveis com exatidão nesse sistema está no intervalo  $[0, 1]$ . É por isso que usualmente deslocamos e normalizamos as entradas de uma imagem digital para esse intervalo, numa tentativa de cometer menos erros de arredondamento ao longo das contas feitas com esses dados.

Em geral, os arquivos de imagens coloridas salvos em formatos comuns como, por exemplo, JPEG e TIFF, ao serem lidos pelo MATLAB, resultam em objetos tridimensionais do tipo `uint8`. Uma função nativa do MATLAB que lê imagens de arquivos é a `imread`. Porque queremos lidar apenas com vídeos em escala de cinza, após a leitura da imagem colorida realizamos a conversão do formato 3D `uint8` para 2D `uint8` usando a função nativa `rgb2gray`, como mencionado anteriormente. A fim de viabilizar as operações matemáticas realizadas pelos métodos de otimização, as imagens digitais precisam ainda ser convertidas para o tipo `double`, já que, do contrário, mesmo as operações mais simples - como a mera soma de matrizes - poderiam resultar em valores impossíveis de serem armazenados com o tipo `uint8`. Essa conversão de

<sup>1</sup> O termo *data class* é adotado por Gonzalez *et. al* para evitar confusão com o que chama de *image type* (binária, indexada, escala de cinza, RGB, etc.), conforme explicado em [43, Seção 1.7.7, *A Note on Terminology*, p. 16].

uma matriz de tipo `uint8` para `double` pode ser feita pela função nativa `im2double`, que para esse tipo de entrada também realiza o reescalamiento dos dados do intervalo  $[0, 255]$  para o intervalo  $[0, 1]$ .

### Exibição de Imagens no MATLAB

Duas funções nativas do MATLAB que podem exibir e salvar imagens digitais são, respectivamente, a `imshow` e `imwrite`, e sua documentação no *help* do programa é suficiente para explorá-las. Também fizemos uso básico da função `export_fig` [1], disponível há mais de 10 anos na página do File Exchange do MATLAB, que facilita o ajuste de certas configurações das imagens a serem salvas em arquivo. Um dos objetivos da `export_fig` é salvar de forma fidedigna a imagem que está sendo exibida no `display`, com qualidade de publicação e impressão. Em última instância, essa função se utiliza das funções padrão (`imwrite`, `print` etc.) para realizar a gravação do arquivo.

Uma situação relevante a ser considerada quando utilizamos métodos de otimização que culminam com a exibição de imagens é aquela em que os valores das entradas das matrizes de imagem fogem do intervalo padrão esperado pelas funções de exibição. Isto porque, nesse contexto, as matrizes geradas são resultado de operações matemáticas que, em geral, podem resultar em quaisquer valores representáveis em ponto flutuante. Quando os valores fogem do intervalo de exibição, algum ajuste necessariamente tem que ser feito, como escalamento (*scaling*) ou projeção dos valores excedentes no extremos intervalares mais próximos (*clipping*).

Segundo Gonzalez e Woods [42, pp .90–91], ao converter imagens para 8 *bits* – que, na prática, é o que se faz para exibir a maioria das imagens – muitos *softwares* adotam o *clipping*, ou seja, transformam os números negativos em 0 e alteram para 255 todos os que ultrapassarem esse valor. É justamente isso que a função `imshow` faz, por predefinição. A escolha automática do intervalo é feita a partir do tipo de dado da imagem:  $[0, 255]$  para `uint8` e  $[0, 1]$  para `double`, para citar os exemplos mais comuns. Alternativamente, o usuário pode fornecer um outro intervalo qualquer como parâmetro de entrada na chamada da função.

Neste trabalho, a decisão que entendemos ser mais coerente com o modelo de otimização quando os valores numéricos excedem o intervalo de exibição, é a de desprezar o sinal dos píxeis numéricos (entradas matriciais) e realizar o escalamento para o intervalo que corresponde à amplitude do vídeo digital em questão.

APÊNDICE F – TABELAS DE RESULTADOS  
FINAIS: VÍDEOS SEM *WARM-START*

Tabela 31 – Resultados dos experimentos numéricos com as instâncias do grupo CAVIAR com ponto de partida trivial para o ADMM e o SPGM.

Instância	m	n	solver	nIter	nSVD	S			L			SL		
						ER	PEP	EMPE	ER	PEP	EMPE	ER	PEP	EMPE
v11	110592	383	ADMM	1000	1000	1.4e-01	0.03	38	6.9e-03	0.03	36	9.7e-03	0.03	37
			SPGM	241	261	4.6e-03	1.51	1	2.2e-04	1.52	1	3.1e-04	1.52	1
v12	110592	750	ADMM	1000	1000	5.8e-02	0.06	12	4.6e-03	0.56	2	6.5e-03	0.31	3
			SPGM	292	339	5.1e-03	1.92	1	4.0e-04	12.89	1	5.7e-04	7.41	1
v13	110592	450	ADMM	1000	1000	5.0e-02	0.06	7	2.6e-03	0.06	6	3.7e-03	0.06	7
			SPGM	236	261	6.3e-03	1.53	1	3.3e-04	1.99	1	4.7e-04	1.76	1
v14	110592	750	ADMM	1000	1000	8.0e-02	0.03	26	5.5e-03	0.05	19	7.8e-03	0.04	22
			SPGM	231	254	3.5e-03	1.60	1	2.4e-04	1.66	1	3.4e-04	1.63	1
v21	110592	750	ADMM	1000	1000	4.4e-02	0.08	6	3.0e-03	0.10	5	4.3e-03	0.09	6
			SPGM	338	370	7.8e-03	3.10	1	5.3e-04	3.55	1	7.5e-04	3.33	1
v22	110592	750	ADMM	1000	1000	5.6e-02	0.05	10	3.6e-03	0.05	9	5.1e-03	0.05	9
			SPGM	283	325	6.3e-03	2.06	1	4.0e-04	3.53	1	5.7e-04	2.80	1
v23	110592	560	ADMM	1000	1000	8.2e-02	0.05	13	4.9e-03	0.05	12	6.9e-03	0.05	13
			SPGM	282	327	6.7e-03	2.16	1	4.0e-04	9.21	1	5.6e-04	5.68	1
v24	110592	750	ADMM	1000	1000	2.7e-03	0.04	1	1.5e-04	0.06	1	2.2e-04	0.05	1
			SPGM	245	280	6.9e-03	2.03	1	3.9e-04	3.06	1	5.5e-04	2.55	1
v25	110592	750	ADMM	1000	1000	1.1e-01	0.06	18	6.4e-03	0.07	14	9.1e-03	0.06	16
			SPGM	318	362	7.3e-03	2.13	1	4.2e-04	3.29	1	5.9e-04	2.71	1
v31	110592	492	ADMM	1000	1000	8.1e-05	0.03	1	4.4e-06	0.03	1	6.2e-06	0.03	1
			SPGM	158	177	3.9e-03	1.89	1	2.1e-04	2.04	1	3.0e-04	1.97	1
v32	110592	750	ADMM	1000	1000	5.5e-03	0.03	2	2.9e-04	0.04	2	4.1e-04	0.04	2
			SPGM	194	212	4.9e-03	1.58	1	2.6e-04	2.79	1	3.6e-04	2.18	1
v33	110592	750	ADMM	1000	1000	7.8e-05	0.02	1	3.9e-06	0.02	1	5.5e-06	0.02	1
			SPGM	184	205	4.2e-03	1.82	1	2.1e-04	3.18	1	2.9e-04	2.50	1
v34	110592	606	ADMM	1000	1000	3.5e-03	0.04	1	2.0e-04	0.05	1	2.8e-04	0.05	1
			SPGM	198	214	4.4e-03	1.84	1	2.5e-04	2.57	1	3.5e-04	2.20	1
v35	110592	959	ADMM	1000	1000	1.6e-04	0.02	1	7.6e-06	0.03	1	1.1e-05	0.03	1
			SPGM	196	218	5.6e-03	1.91	1	2.7e-04	2.18	1	3.8e-04	2.04	1

Tabela 32 – Resultados dos experimentos numéricos com as instâncias do grupo SBI com ponto de partida trivial para o ADMM e o SPGM.

Instância	m	n	solver	nIter	nSVD	S			L			SL		
						ER	PEP	EMPE	ER	PEP	EMPE	ER	PEP	EMPE
Board	32800	228	ADMM	1000	1000	3.3e-05	0.04	1	5.0e-06	0.04	1	7.0e-06	0.04	1
			SPGM	395	437	4.0e-03	5.19	1	6.0e-04	7.18	1	8.4e-04	6.19	1
Candela_m1.10	101376	350	ADMM	1000	1000	2.2e-04	0.06	1	1.4e-05	0.09	1	1.9e-05	0.07	1
			SPGM	227	252	5.6e-03	1.49	1	3.4e-04	1.34	1	4.9e-04	1.42	1
CAVIAR1	98304	610	ADMM	1000	1000	5.9e-03	0.81	1	3.7e-04	0.78	1	5.3e-04	0.79	1
			SPGM	283	328	1.4e-02	3.66	1	8.7e-04	3.76	1	1.2e-03	3.71	1
CAVIAR2	98304	460	ADMM	1000	1000	9.9e-05	0.03	1	4.8e-06	0.04	1	6.7e-06	0.03	1
			SPGM	180	203	4.0e-03	1.51	1	2.0e-04	4.08	1	2.8e-04	2.80	1
CaVignal	27200	258	ADMM	1000	1000	1.6e-04	0.10	1	1.6e-05	0.91	1	2.2e-05	0.50	1
			SPGM	210	237	3.2e-03	1.78	1	3.2e-04	13.08	1	4.5e-04	7.43	1
Foliage	28800	394	ADMM	1000	1000	1.3e-05	0.01	1	2.4e-06	0.01	1	3.4e-06	0.01	1
			SPGM	392	449	3.8e-03	5.76	1	6.9e-04	5.58	1	9.6e-04	5.67	1
HallAndMonitor	84480	296	ADMM	1000	1000	1.7e-02	0.16	3	1.4e-03	2.09	1	2.0e-03	1.12	1
			SPGM	216	246	5.5e-03	2.96	1	4.6e-04	3.34	1	6.5e-04	3.15	1
HighwayI	76800	440	ADMM	1000	1000	2.2e-05	0.05	1	3.9e-06	0.02	1	5.4e-06	0.03	1
			SPGM	316	350	3.2e-03	7.48	1	5.9e-04	8.22	1	8.2e-04	7.85	1
HighwayII	76800	500	ADMM	1000	1000	3.5e-05	0.00	1	4.8e-06	0.00	1	6.8e-06	0.00	1
			SPGM	180	193	1.7e-03	2.64	1	2.4e-04	2.34	1	3.4e-04	2.49	1
HumanBody2	76800	740	ADMM	1000	1000	2.4e-04	0.05	1	3.0e-05	0.12	1	4.3e-05	0.09	1
			SPGM	261	297	3.7e-03	3.11	1	4.7e-04	12.89	1	6.6e-04	8.00	1
IBMtest2	76800	90	ADMM	1000	1000	3.1e-05	0.02	1	4.6e-06	0.02	1	6.5e-06	0.02	1
			SPGM	175	197	2.1e-03	3.10	1	3.1e-04	3.69	1	4.3e-04	3.40	1
PeopleAndFoliage	76800	341	ADMM	1000	1000	3.1e-03	1.21	1	7.4e-04	1.65	1	1.0e-03	1.43	1
			SPGM	418	478	3.1e-03	5.17	1	7.5e-04	5.09	1	1.0e-03	5.13	1
Snellen	20736	321	ADMM	1000	1000	1.2e-04	0.08	1	1.3e-05	0.05	1	1.8e-05	0.06	1
			SPGM	325	378	6.1e-03	9.03	1	6.5e-04	7.25	1	9.1e-04	8.14	1
Toscana	480000	6	ADMM	1000	1000	1.4e-01	44.99	4	3.7e-02	81.70	2	5.0e-02	63.35	3
			SPGM	584	666	3.2e-03	6.73	1	8.1e-04	6.73	1	1.1e-03	6.73	1

APÊNDICE G – TABELAS DE RESULTADOS  
FINAIS: VÍDEOS COM *WARM-START*

Tabela 33 – Resultados dos experimentos numéricos com as instâncias do grupo CAVIAR com a heurística de warm-start para o ADMM e o SPGM.

Instância	m	n	solver	nIter	nSVD	S			L			SL		
						ER	PEP	EMPE	ER	PEP	EMPE	ER	PEP	EMPE
v11	110592	383	ADMM+	1000	1000	2.9e-02	0.02	9	1.4e-03	0.02	9	2.0e-03	0.02	9
			SPGM+	226	242	4.7e-03	1.45	1	2.3e-04	1.47	1	3.2e-04	1.46	1
v12	110592	750	ADMM+	1000	1000	2.8e-02	0.03	8	2.2e-03	0.52	1	3.1e-03	0.27	2
			SPGM+	299	343	5.1e-03	1.94	1	4.0e-04	13.01	1	5.7e-04	7.47	1
v13	110592	450	ADMM+	1000	1000	9.0e-03	0.03	3	4.7e-04	0.03	3	6.7e-04	0.03	3
			SPGM+	240	259	6.2e-03	1.42	1	3.2e-04	1.84	1	4.6e-04	1.63	1
v14	110592	750	ADMM+	1000	1000	1.5e-03	0.02	1	1.1e-04	0.02	1	1.5e-04	0.02	1
			SPGM+	215	228	3.2e-03	1.53	1	2.2e-04	1.59	1	3.2e-04	1.56	1
v21	110592	750	ADMM+	1000	1000	1.1e-03	0.02	1	7.3e-05	0.05	1	1.0e-04	0.04	1
			SPGM+	333	365	7.4e-03	3.04	1	5.1e-04	3.48	1	7.2e-04	3.26	1
v22	110592	750	ADMM+	1000	1000	1.9e-02	0.02	5	1.2e-03	0.06	3	1.7e-03	0.04	4
			SPGM+	295	321	6.6e-03	2.22	1	4.2e-04	3.79	1	5.9e-04	3.01	1
v23	110592	560	ADMM+	1000	1000	1.3e-02	0.03	3	7.7e-04	0.06	2	1.1e-03	0.05	2
			SPGM+	280	314	6.8e-03	2.17	1	4.0e-04	9.22	1	5.7e-04	5.69	1
v24	110592	750	ADMM+	1000	1000	9.8e-05	0.02	1	5.6e-06	0.03	1	7.9e-06	0.03	1
			SPGM+	259	286	6.0e-03	1.89	1	3.4e-04	2.89	1	4.9e-04	2.39	1
v25	110592	750	ADMM+	1000	1000	2.8e-02	0.03	6	1.6e-03	0.29	2	2.3e-03	0.16	2
			SPGM+	298	337	7.7e-03	2.25	1	4.4e-04	3.40	1	6.3e-04	2.82	1
v31	110592	492	ADMM+	1000	1000	6.5e-05	0.02	1	3.5e-06	0.02	1	5.0e-06	0.02	1
			SPGM+	162	175	3.6e-03	1.68	1	1.9e-04	1.85	1	2.7e-04	1.76	1
v32	110592	750	ADMM+	1000	1000	6.1e-03	0.03	2	3.2e-04	0.03	1	4.5e-04	0.03	1
			SPGM+	184	196	5.2e-03	1.75	1	2.7e-04	3.06	1	3.8e-04	2.41	1
v33	110592	750	ADMM+	1000	1000	6.4e-05	0.02	1	3.2e-06	0.02	1	4.5e-06	0.02	1
			SPGM+	174	183	4.6e-03	1.96	1	2.3e-04	3.43	1	3.2e-04	2.70	1
v34	110592	606	ADMM+	1000	1000	1.3e-03	0.02	1	7.4e-05	0.03	1	1.0e-04	0.03	1
			SPGM+	192	204	4.4e-03	1.82	1	2.5e-04	2.56	1	3.5e-04	2.19	1
v35	110592	959	ADMM+	1000	1000	6.2e-05	0.02	1	3.0e-06	0.02	1	4.2e-06	0.02	1
			SPGM+	195	212	5.6e-03	1.94	1	2.7e-04	2.21	1	3.8e-04	2.08	1

Tabela 34 – Resultados dos experimentos numéricos com as instâncias do grupo SBI com a heurística de *warm-start* para o ADMM e o SPGM.

Instância	m	n	solver	nIter	nSVD	S			L			SL		
						ER	PEP	EMPE	ER	PEP	EMPE	ER	PEP	EMPE
Board	32800	228	ADMM+	1000	1000	2.8e-05	0.03	1	4.3e-06	0.03	1	6.0e-06	0.03	1
			SPGM+	352	390	4.2e-03	5.41	1	6.4e-04	7.41	1	8.9e-04	6.41	1
Candela_m1.10	101376	350	ADMM+	1000	1000	1.2e-04	0.05	1	7.7e-06	0.07	1	1.1e-05	0.06	1
			SPGM+	239	256	5.0e-03	1.45	1	3.1e-04	1.30	1	4.3e-04	1.38	1
CAVIAR1	98304	610	ADMM+	1000	1000	5.9e-03	0.81	1	3.7e-04	0.79	1	5.3e-04	0.80	1
			SPGM+	285	315	1.4e-02	3.57	1	8.8e-04	3.67	1	1.2e-03	3.62	1
CAVIAR2	98304	460	ADMM+	1000	1000	1.1e-04	0.03	1	5.4e-06	0.04	1	7.6e-06	0.04	1
			SPGM+	183	194	3.9e-03	1.43	1	1.9e-04	3.69	1	2.6e-04	2.56	1
CaVignal	27200	258	ADMM+	1000	1000	1.5e-04	0.09	1	1.5e-05	0.87	1	2.1e-05	0.48	1
			SPGM+	241	266	3.3e-03	1.85	1	3.3e-04	13.68	1	4.6e-04	7.77	1
Foliage	28800	394	ADMM+	1000	1000	1.1e-05	0.01	1	2.0e-06	0.01	1	2.8e-06	0.01	1
			SPGM+	365	415	3.9e-03	5.93	1	7.1e-04	5.72	1	9.9e-04	5.83	1
HallAndMonitor	84480	296	ADMM+	1000	1000	4.1e-03	0.05	2	3.5e-04	0.19	1	4.9e-04	0.12	1
			SPGM+	244	269	4.9e-03	2.59	1	4.1e-04	2.80	1	5.8e-04	2.69	1
Highway1	76800	440	ADMM+	1000	1000	2.3e-05	0.05	1	4.2e-06	0.02	1	5.9e-06	0.03	1
			SPGM+	305	339	3.2e-03	7.41	1	5.8e-04	7.94	1	8.0e-04	7.68	1
HighwayII	76800	500	ADMM+	1000	1000	4.4e-05	0.00	1	6.1e-06	0.00	1	8.5e-06	0.00	1
			SPGM+	182	191	1.7e-03	2.60	1	2.4e-04	2.31	1	3.4e-04	2.46	1
HumanBody2	76800	740	ADMM+	1000	1000	8.2e-05	0.02	1	1.0e-05	0.04	1	1.5e-05	0.03	1
			SPGM+	289	322	3.8e-03	3.15	1	4.8e-04	13.86	1	6.8e-04	8.50	1
IBMtest2	76800	90	ADMM+	1000	1000	2.8e-05	0.01	1	4.1e-06	0.02	1	5.7e-06	0.01	1
			SPGM+	168	187	2.1e-03	3.07	1	3.1e-04	3.62	1	4.3e-04	3.34	1
PeopleAndFoliage	76800	341	ADMM+	1000	1000	3.7e-04	0.14	1	8.9e-05	0.15	1	1.2e-04	0.14	1
			SPGM+	429	493	3.0e-03	4.86	1	7.1e-04	4.76	1	9.8e-04	4.81	1
Snellen	20736	321	ADMM+	1000	1000	1.1e-04	0.07	1	1.2e-05	0.04	1	1.7e-05	0.05	1
			SPGM+	344	382	5.8e-03	8.57	1	6.2e-04	6.91	1	8.7e-04	7.74	1
Toscana	480000	6	ADMM+	1000	1000	2.4e-02	7.08	3	6.1e-03	6.74	2	8.3e-03	6.91	2
			SPGM+	534	608	3.5e-03	7.19	1	9.0e-04	7.09	1	1.2e-03	7.14	1

APÊNDICE H – TABELAS DE RESULTADOS  
INICIAIS DOS VÍDEOS COM E SEM  
*WARM-START*

Tabela 35 – Dados comparativos dos pontos iniciais do grupo CAVIAR com e sem a heurística de *warm-start*. A indicação da partida aprimorada é feita por meio do símbolo “+” no rótulo da coluna.

Instância	solteir	rnk		spr		spr		ER		PEP		EMPE				
		$L^{(0)+}$	$L^{*+}$	$S^{(0)}$	$S^{(0)+}$	$S_V^{(0)}$	$S_V^{(0)+}$	$S^{*+}$	$L^{(0)}$	$L^{(0)+}$	$L^{(0)}$	$L^{(0)+}$	$L^{(0)}$	$L^{(0)+}$		
v11	ADMM	1	184	99.6	100	99.6	100	99.6	80.5	39.4	1.4e+00	3.3e-02	99.7	80.9	133	2
v11	SPGM	1	176	99.6	100	99.6	100	99.6	80.5	40	1.4e+00	3.3e-02	99.7	80.9	133	2
v12	ADMM	2	368	99.4	100	99.4	100	99.4	88	39.8	1.4e+00	6.7e-02	99.5	98.9	130	14
v12	SPGM	2	350	99.4	100	99.4	100	99.4	88	40.3	1.4e+00	6.7e-02	99.5	98.9	130	14
v13	ADMM	1	216	99.6	100	99.6	100	99.6	80.8	39.8	1.4e+00	3.6e-02	99.7	81.2	133	3
v13	SPGM	1	209	99.6	100	99.6	100	99.6	80.8	40.3	1.4e+00	3.6e-02	99.7	81.2	133	3
v14	ADMM	2	366	99.6	100	99.6	100	99.6	83.6	38.3	1.4e+00	4.8e-02	99.7	99.3	133	15
v14	SPGM	2	350	99.6	100	99.6	100	99.6	83.6	38.9	1.4e+00	4.8e-02	99.7	99.3	133	15
v21	ADMM	37	370	99.5	100	99.5	100	99.5	82.4	45.6	1.4e+00	6.1e-02	99.5	99.2	137	35
v21	SPGM	37	355	99.5	100	99.5	100	99.5	82.4	45.9	1.4e+00	6.1e-02	99.5	99.2	137	35
v22	ADMM	3	376	99.5	100	99.5	100	99.5	88	43.9	1.4e+00	8.4e-02	99.5	99.2	141	29
v22	SPGM	3	359	99.5	100	99.5	100	99.5	88	44.4	1.4e+00	8.4e-02	99.5	99.2	141	29
v23	ADMM	13	278	99.5	100	99.5	100	99.5	77.4	43.3	1.4e+00	5.8e-02	99.5	99.3	139	37
v23	SPGM	13	263	99.5	100	99.5	100	99.5	77.4	43.9	1.4e+00	5.8e-02	99.5	99.3	139	37
v24	ADMM	14	380	99.5	100	99.5	100	99.5	78	44.1	1.4e+00	5.9e-02	99.5	99.3	140	41
v24	SPGM	14	362	99.5	100	99.5	100	99.5	78	44.6	1.4e+00	5.9e-02	99.5	99.3	140	41
v25	ADMM	13	378	99.5	100	99.5	100	99.5	76.7	43.8	1.4e+00	5.3e-02	99.5	99.2	138	33
v25	SPGM	13	361	99.5	100	99.5	100	99.5	76.7	44.4	1.4e+00	5.3e-02	99.5	99.2	138	33
v31	ADMM	1	223	100	100	100	100	100	88.5	60.1	1.4e+00	3.6e-02	100	83.9	122	3
v31	SPGM	1	213	100	100	100	100	100	88.5	60.3	1.4e+00	3.6e-02	100	83.9	122	3
v32	ADMM	1	362	99.2	100	99.2	100	99.2	83.8	59.2	1.4e+00	3.1e-02	99.2	88.1	123	3
v32	SPGM	1	330	99.2	100	99.2	100	99.2	83.8	59.6	1.4e+00	3.1e-02	99.2	88.1	123	3
v33	ADMM	2	347	99.4	100	99.4	100	99.4	83.3	60.7	1.4e+00	3.1e-02	99.4	88.8	132	3
v33	SPGM	2	328	99.4	100	99.4	100	99.4	83.3	61.1	1.4e+00	3.1e-02	99.4	88.8	132	3
v34	ADMM	1	280	99	100	99	100	99	84.6	60.2	1.4e+00	3.4e-02	99	87	129	3
v34	SPGM	1	262	99	100	99	100	99	84.6	60.5	1.4e+00	3.4e-02	99	87	129	3
v35	ADMM	4	514	100	100	100	100	100	84.6	58.6	1.4e+00	3.5e-02	100	99.7	122	13
v35	SPGM	4	474	100	100	100	100	100	84.6	59	1.4e+00	3.5e-02	100	99.7	122	13

Tabela 36 – Dados comparativos dos pontos iniciais do grupo SBI com e sem a heurística de *warm-start*. A indicação da partida aprimorada é feita por meio do símbolo “+” no rótulo da coluna.

Instância	soltoer	rnk		spr		spr		spr		ER		PEP		EMPE	
		$L^{(0)+}$	$L^{**}$	$S^{(0)}$	$S^{(0)+}$	$S_V^{(0)}$	$S_V^{(0)+}$	$S_V^{(0)}$	$S_V^{(0)+}$	$L^{(0)}$	$L^{(0)+}$	$L^{(0)}$	$L^{(0)+}$	$L^{(0)}$	$L^{(0)+}$
Board	ADMM	12	119	100	100	100	97.2	58.5	1.4e+00	1.1e-01	100	97.2	134	11	
Board	SPGM	12	117	100	100	100	97.2	58.7	1.4e+00	1.1e-01	100	97.2	134	11	
Candela-m1.10	ADMM	2	160	100	100	100	75.4	37.4	1.4e+00	3.7e-02	100	99.4	126	6	
Candela-m1.10	SPGM	2	134	100	100	100	75.4	37.6	1.4e+00	3.7e-02	100	99.4	126	6	
CAVIAR1	ADMM	9	296	100	100	100	86.3	48.3	1.4e+00	6.3e-02	100	95	127	8	
CAVIAR1	SPGM	9	280	100	100	100	86.3	48.8	1.4e+00	6.3e-02	100	95	127	8	
CAVIAR2	ADMM	1	215	100	100	100	81.7	44.7	1.4e+00	2.5e-02	100	95	120	4	
CAVIAR2	SPGM	1	206	100	100	100	81.7	45.2	1.4e+00	2.5e-02	100	95	120	4	
CaVignal	ADMM	4	53	100	100	100	63	16.3	1.4e+00	8.3e-02	100	99.6	163	22	
CaVignal	SPGM	4	48	100	100	100	63	16.4	1.4e+00	8.3e-02	100	99.6	163	22	
Foliage	ADMM	24	209	100	100	100	97.8	72.2	1.4e+00	1.5e-01	100	98.6	91	18	
Foliage	SPGM	24	209	100	100	100	97.8	72.4	1.4e+00	1.5e-01	100	98.6	91	18	
HallAndMonitor	ADMM	2	117	99.9	100	100	99.9	87	1.4e+00	6.0e-02	100	99.5	145	26	
HallAndMonitor	SPGM	2	111	99.9	100	100	99.9	87	1.4e+00	6.0e-02	100	99.5	145	26	
Highway1	ADMM	7	217	100	100	100	96.7	76.1	1.4e+00	1.2e-01	100	99.5	148	23	
Highway1	SPGM	7	214	100	100	100	96.7	76.4	1.4e+00	1.2e-01	100	99.5	148	23	
HighwayII	ADMM	1	289	100	100	100	94.3	82.1	1.4e+00	5.7e-02	100	94.7	121	5	
HighwayII	SPGM	1	275	100	100	100	94.3	82.5	1.4e+00	5.7e-02	100	94.7	121	5	
HumanBody2	ADMM	12	353	100	100	100	92.4	59.4	1.4e+00	1.1e-01	100	99.1	113	26	
HumanBody2	SPGM	12	340	100	100	100	92.4	59.7	1.4e+00	1.1e-01	100	99.1	113	26	
IBMtest2	ADMM	1	24	99.4	100	100	99.4	94.2	81	1.4e+00	6.4e-02	99.7	93.2	143	6
IBMtest2	SPGM	1	23	99.4	100	100	99.4	94.2	81.1	1.4e+00	6.4e-02	99.7	93.2	143	6
PeopleAndFoliage	ADMM	9	175	100	100	100	96.9	66.6	1.4e+00	1.8e-01	100	97.9	89	19	
PeopleAndFoliage	SPGM	9	172	100	100	100	96.9	66.7	1.4e+00	1.8e-01	100	97.9	89	19	
Snellen	ADMM	30	173	99.9	100	100	99.9	98.1	70.9	1.4e+00	9.2e-02	100	96.2	127	8
Snellen	SPGM	30	172	99.9	100	100	99.9	98.1	71.2	1.4e+00	9.2e-02	100	96.2	127	8
Toscana	ADMM	1	3	100	100	100	98.3	74.9	1.4e+00	1.7e-01	100	98.8	108	14	
Toscana	SPGM	1	3	100	100	100	98.3	75	1.4e+00	1.7e-01	100	98.8	108	14	

APÊNDICE I – GRÁFICOS COMPARATIVOS  
DOS RESULTADOS DAS INSTÂNCIAS DOS  
VÍDEOS COM *WARM-START* (ER,  
PEP,EMPE)

Figura 30 – **Warm-Start**. Erro Relativo (ER), Erro em Percentual de Pixels Errados (PEP) e Erro Médio por Pixel Errado (EMPE) para cada uma das instâncias dos vídeos do Grupo CAVIAR.

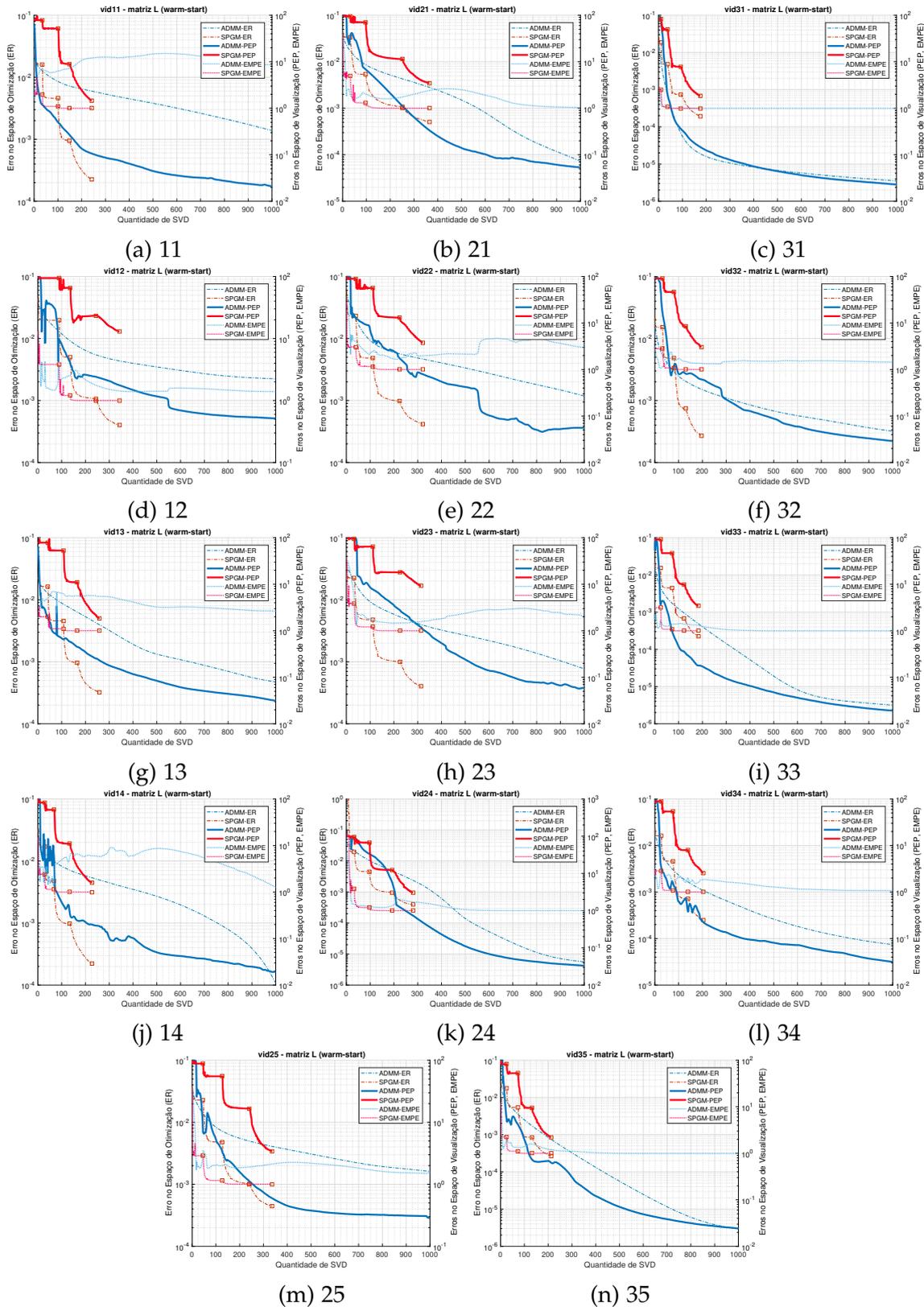
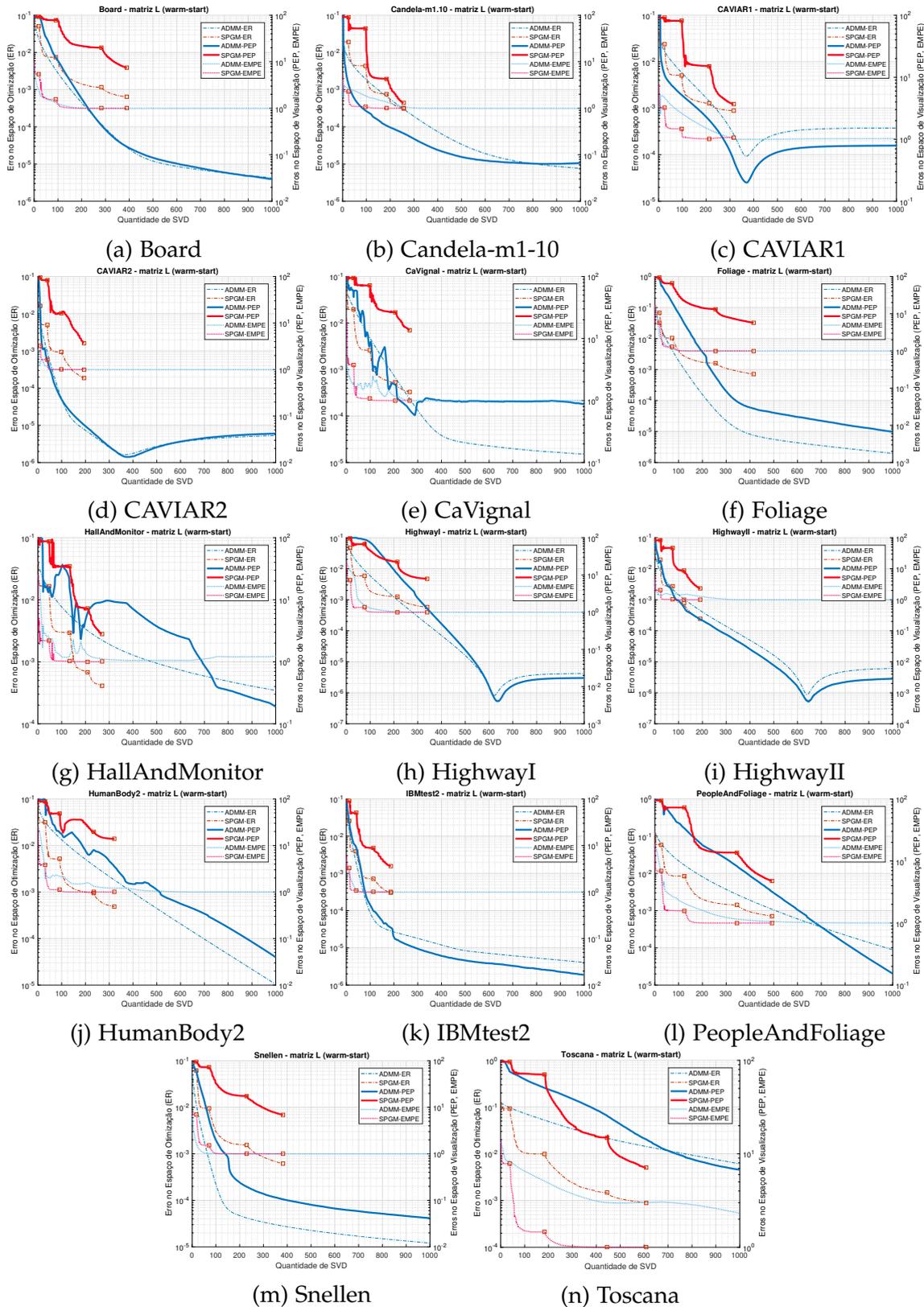


Figura 31 – *Warm-Start*. Erro Relativo (ER), Erro em Percentual de Pixels Errados (PEP) e Erro Médio por Pixel Errado (EMPE) para cada uma das instâncias dos vídeos do Grupo SBI.



## Anexos

# ANEXO A – TABELAS PUBLICADAS COM RESULTADOS DO ADMM

Tabela 37 – Tabela reproduzida e adaptada de [103, Tabela 1], com tradução livre.  
“Resultados recuperados:  $A^*$  é esparsa impulsiva”.

$m = n$	$t$	$r$	spr	ErrSP	ErrLR	RelErr	nIter	Tempo (s)
100	0.1	10	5%	5.8e-5	2.4e-4	1.6e-4	13	0.12
			10%	1.4e-4	3.1e-4	2.1e-4	17	0.17
500	0.05	10	5%	3.3e-5	1.9e-5	2.9e-5	10	2.7
			10%	5.0e-5	5.1e-5	5.0e-5	11	2.8
500	0.05	50	5%	4.4e-5	2.7e-4	3.8e-5	13	3.9
			10%	5.2e-5	5.8e-5	5.4e-5	15	3.8
1000	0.05	50	5%	2.8e-5	1.8e-5	2.4e-5	12	23
			10%	5.5e-5	5.4e-5	5.5e-5	13	25
1000	0.05	100	5%	4.6e-5	3.6e-5	4.2e-5	14	27
			10%	7.7e-5	7.4e-5	7.6e-5	16	31

Tabela 38 – Tabela reproduzida e adaptada de [103, Tabela 2], com tradução livre.  
“Resultados recuperados:  $A^*$  é esparsa Gaussiana”.

$m = n$	$t$	$r$	spr	ErrSP	ErrLR	RelErr	nIter	Tempo (s)
100	0.1	10	5%	3.5e-2	1.3e-3	2.9e-3	48	0.44
			10%	1.5e-2	8.3e-4	1.8e-3	82	0.77
500	0.05	10	5%	6.2e-3	9.3e-5	4.5e-4	55	14
			10%	6.4e-3	1.5e-4	6.7e-4	57	15
500	0.05	50	5%	2.2e-2	3.3e-4	7.7e-4	71	18
			10%	1.7e-2	3.8e-4	8.4e-4	89	23
1000	0.05	50	5%	1.8e-2	2.0e-4	6.1e-4	69	135
			10%	1.6e-2	3.2e-4	7.7e-4	82	165
1000	0.05	100	5%	2.8e-2	3.0e-4	7.0e-4	84	165
			10%	2.5e-2	5.0e-4	9.3e-4	107	212

Tabela 39 – Tabela reproduzida e adaptada de [21, Tabela I], com tradução livre. “Recuperação Correta para Problemas Aleatórios com Variação de Tamanho”.

Dimensão $n$	$\text{rnk}(L^*)$	$\text{zeros}(S^*)$	$\text{rnk}(L)$	$\text{zeros}(S)$	ErrLR	nSVD	Tempo (s)
500	25	12,500	25	12,500	1.1e-6	16	2.9
1000	50	50000	50	50000	1.2e-6	16	12.4
2000	100	200000	100	200000	1.2e-6	16	61.8
3000	250	450000	250	450000	2.3e-6	15	185.2

$$\text{rnk}(L^*) = 0.05 \times n, \text{ zeros}(S^*) = 0.05 \times n^2.$$

Dimensão $n$	$\text{rnk}(L^*)$	$\text{zeros}(S^*)$	$\text{rnk}(L)$	$\text{zeros}(S)$	ErrLR	nSVD	Tempo (s)
500	25	25000	25	25000	1.2e-6	17	4.0
1000	50	100000	50	100000	2.4e-6	16	13.7
2000	100	400000	100	400000	2.4e-6	16	64.5
3000	150	900000	150	900000	2.5e-6	16	191.0

$$\text{rank}(L^*) = 0.05 \times n, \text{ zeros}(S^*) = 0.10 \times n^2.$$