UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Elétrica e de Computação

OSCAR ERNESTO ROJAS ROJAS

DETECÇÃO AUTOMÁTICA DE COMPORTAMENTOS ANORMAIS EM VÍDEOS DE SISTEMAS DE VIGILÂNCIA E MONITORAMENTO

Campinas

2017

OSCAR ERNESTO ROJAS ROJAS

DETECÇÃO AUTOMÁTICA DE COMPORTAMENTOS ANORMAIS EM VÍDEOS DE SISTEMAS DE VIGILÂNCIA E MONITORAMENTO

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Doutor em Engenharia Elétrica, na Área de Engenharia de Computação.

Orientador: Prof. Dr. Clesio Luiz Tozzi

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA TESE DEFENDIDA PELO ALUNO OSCAR ERNESTO ROJAS ROJAS, E ORIENTADA PELO PROF. DR. CLESIO LUIZ TOZZI

Campinas

2017

Agência(s) de fomento e nº(s) de processo(s): CNPq, 141867/2013-2
ORCID: http://orcid.org/0000-0002-1867-7445

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Luciana Pietrosanto Milla - CRB 8/8129

R638d

Rojas Rojas, Oscar Ernesto, 1987-
Detecção automática de comportamentos anormais em vídeos de sistemas de vigilância e monitoramento / Oscar Ernesto Rojas Rojas. – Campinas, SP : [s.n.], 2017.

Orientador: Clesio Luis Tozzi.
Tese (doutorado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Reconhecimento de padrões. 2. Processamento de imagens. 3. Visão artificial. 4. Fluxo óptico. 5. Sistemas inteligentes de controle. I. Tozzi, Clesio Luis,1948-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Automatic detection of abnormal behaviors in video surveillance and monitoring systems
Palavras-chave em inglês:
Pattern recognition
Image processing
Artificial vision
Optical flow
Intelligent control systems
Área de concentração: Automação
Titulação: Doutor em Engenharia Elétrica
Banca examinadora:
Clesio Luis Tozzi [Orientador]
João Paulo Papa
Arthur de Miranda Neto
Leticia Rittner
Ricardo Ribeiro Gudwin
Data de defesa: 07-07-2017
Programa de Pós-Graduação: Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE DOUTORADO

Candidato: Oscar Ernesto Rojas Rojas. RA: 115679

Data da Defesa: 7 de Julho de 2017

Título da Tese: Detecção Automática de Comportamentos Anormais em Vídeos de Sistemas de Vigilância e Monitoramento.

Prof. Dr. Clesio Luis Tozzi (Presidente, FEEC/UNICAMP)

Prof. Dr. João Paulo Papa (UNESP)

Prof. Dr. Arthur de Miranda Neto (DEG/UFLA)

Profa. Dra. Leticia Rittner (FEEC/UNICAMP)

Prof. Dr. Ricardo Ribeiro Gudwin (FEEC/UNICAMP)

A ata de defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no processo de vida acadêmica do aluno.

Dedico esta tese a minha família,
obrigado por fazer de mim quem eu sou.

Agradecimentos

Agradeço primeiramente a Deus pela oportunidade de realizar este trabalho.

Ao meu orientador Prof. Dr. Clésio Luis Tozzi, pela oportunidade de desenvolver este trabalho sob sua orientação.

A minha família: minha mãe e minha irmã pela paciência, compreensão e apoio.

Aos meus amigos: Diana Martinez, German Castañeda, Suranny Jimenez, Ramiro Chamorro, Gilberto Luis Valente, Angie Forero, Andres Puerto, Sabrina, Sebastian Chavez e a todos aqueles com os que compartilhei diversos momentos.

Agradeço especialmente a Paola Gonzalez Ramos pelo apoio, amizade, compreensão, paciência e carinho em todos os momentos compartilhados. Ajudou-me muito nesse período e tudo teria sido muito mais difícil sem a sua presença. Muito obrigado.

A Juliana Momoe e Miriam Dominguez por sua valiosa companhia e momentos de muita alegria, lembrarei de vocês sempre.

A todas as pessoas que fizeram parte desse período e fizeram de minha estada no Brasil uma experiência muito boa.

Ao CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico - pelo indispensável apoio financeiro.

"Curiosity is the very basis of education
and if you tell me that curiosity killed the cat, I say only the cat died nobly."
(Arnold Edinborough)

Resumo

Nos sistemas de segurança atuais são usadas câmeras para auxiliar no monitoramento de espaços como aeroportos, prédios, praças, parques, entre outros. Muitos desses sistemas utilizam dezenas ou até centenas de câmeras que geram imagens que devem ser analisadas por operadores. Com uma alta quantidade de câmeras o número de operadores necessários para analisar as imagens aumenta, tornando o sistema mais custoso, ainda por cima, nos sistemas atuais um único operador deve analisar dezenas de imagens simultaneamente durante longos períodos de tempo possibilitando que, devido ao cansaço ou por distrações, anormalidades passem despercebidas. Visando reduzir o número de operadores necessários assim como a quantidade de atenção requerida por parte deles para analisar as imagens, neste trabalho é proposto, implementado e testado um sistema de detecção automático de um conjunto preestabelecido de quatro casos de comportamentos considerados anormais relativos ao monitoramento de grupos de pessoas. O primeiro caso se relaciona ao aumento repentino na velocidade das pessoas em uma cena, por exemplo, quando elas estão inicialmente caminhando e, por alguma razão, de repente começam a correr. O segundo comportamento associa-se às mudanças nos padrões de posição, velocidade ou direção de deslocamento de uma ou várias pessoas na cena. No terceiro caso de anormalidade um indivíduo tenta atravessar uma multidão deslocando-se em direção contrária à direção de movimento desta. O quarto e último caso de anormalidade se relaciona à presença de objetos com baixa probabilidade de aparição segundo o contexto da cena, por exemplo, carros numa travessia de pedestres ou pessoas atravessando uma rodovia a pé. As três primeiras anormalidades são baseadas em padrões de movimento, enquanto a quarta anormalidade é baseada na aparência dos objetos.

As técnicas de detecção de anormalidade propostas neste trabalho utilizam ferramentas de visão computacional e processamento digital de imagens e cada uma delas é dividida em duas etapas: na primeira etapa, ou etapa de treinamento, são extraídas as características principais de movimento e aparência (*features*) das imagens de entrada que são usadas para gerar um modelo de comportamento normal. Na segunda etapa, ou etapa de teste, ditas características são obtidas de novas imagens e são comparadas com o modelo gerado anteriormente classificando-as como contendo ou não anormalidade. Os algoritmos desenvolvidos foram projetados para garantir modelos atualizáveis, sendo reunidos em uma arquitetura baseada em módulos, que permite tratar cada caso de anormalidade de forma independente, e permite incluir, futuramente, algoritmos para detecção de anormalidades não contempladas neste trabalho. A arquitetura por módulos também permite obter mais informações em relação ao tipo de anormalidade do que as reportadas nos trabalhos correlatos.

O sistema proposto foi avaliado utilizando conjuntos de vídeos publicamente disponíveis

e os resultados foram comparados com trabalhos correlatos demonstrando desempenho satisfatório, tanto na detecção quanto na localização das situações aqui tratadas.

Palavras-chaves: Detecção de comportamento anormal; Sistemas inteligentes de vigilância; Visão computacional; Processamento de imagens.

Abstract

Current surveillance systems use video cameras to monitoring locations as airports, buildings, parks, among others. Many of those systems require dozens or even hundreds of cameras, each one producing images that are analyzed by operators. With the increasing number of cameras, the number of operators required to analyze all the data increase as well, producing expensive systems. Moreover, in nowadays systems a single operator must analyze dozens of images simultaneously over long periods of time, leading the possibility of unnoticed abnormalities due to fatigue or distractions. In order to reduce the number of operators required, as well as the amount of attention that they required to analyze the images, in this work we propose, implement and test a system for automatic detection of a pre-established set of four types of behaviors considered as abnormals related to the monitoring of groups of people. The first type of abnormal behavior is related to the sudden increase of people's speed in a scene. It occurs for example when they are initially walking and, for some reason, suddenly they start to run. The second type of behavior is associated with changes in the patterns of position, speed and direction of displacement of one or more persons in the scene. In the third type, a person tries to walk across a crowd by moving in the opposite direction to the crowd. The fourth and final type of abnormality is related to the presence of objects with a low probability of occurrence according to the scene's context, for example, cars in a pedestrian crossing area. The first three abnormalities are based on motion patterns, while the fourth is based on the object's appearance.

The techniques of abnormality detection proposed in this work use computer vision and digital image processing tools and each one is divided into two stages: in the first stage, called the training stage, features of movement and appearance are extracted and used to build a model of normal behavior. In the second stage, or test stage, such features are extracted from new images and are compared against the previously generated model classifying them as presenting or not an abnormality. The developed algorithms were designed to guarantee updatable models and were assembled in a module-based architecture, which permits treating each type of abnormality independently, allowing to include, in the future, algorithms for abnormalities detection not contemplated in this work. The architecture by modules allows obtain more information regarding the type of abnormality than those reported in the related works.

The proposed system was evaluated using publicly available datasets and the results were compared with related works showing satisfactory performance in both detection and localization of the treated situations.

Keywords: Abnormal behavior detection; Intelligent surveillance systems; Computer vision; Image processing.

Lista de ilustrações

Lista de tabelas

Sumário

CAPÍTULO 1

Introdução

Atualmente, ano de 2017, no mundo existem 7,4 bilhões de pessoas e o número continua aumentando de forma exponencial:

> - Pense no seguinte: a população da Terra levou milhares de anos, desde a aurora da humanidade até o início do século XIX, para atingir um bilhão de pessoas. Então, de forma estarrecedora, precisou apenas de uns cem anos para duplicar e chegar a dois bilhões, na década de 1920. Depois disso, em meros cinquenta anos, a população tornou a duplicar para quatro bilhões na década de 1970. (...) Só hoje a raça humana acrescentou outras 250 mil pessoas ao planeta Terra. Um quarto de milhão. E isso acontece todos os dias. Atualmente, a cada ano, acrescentamos ao planeta um pouco mais que o equivalente a toda a população da Alemanha.[1]

A Organização das Nações Unidas (ONU) estimou que, para o ano 2050, o número de habitantes no mundo poderia atingir 9,2 bilhões de pessoas. O aumento da população demanda um crescimento das cidades e, consequentemente, novos desafios em matéria de segurança pública e sistemas de vigilância. Atualmente nas grandes cidades estão sendo utilizadas câmeras de vídeo instaladas em pontos estratégicos para identificar situações de risco ou registrar sujeitos que cometeram algum delito. As melhorias nas câmeras de vídeo, nos computadores, nos sistemas de transmissão e na internet fizeram dos sistemas de vigilância por vídeo uma forma muito popular de monitorar um espaço onde uma atividade de interesse pode acontecer. Atualmente podem-se enumerar dois cenários onde as câmeras de vídeo são utilizadas para vigilância: o primeiro, chamado aqui de sistema passivo, é aquele no qual a câmera permanece gravando um local sem supervisão humana, por exemplo uma loja ou um banco durante a noite; nesse cenário as imagens são recuperadas

[1] Inferno, Dan Brown, p. 101, Editorial Arqueiro, 2013

e analisadas somente se algum evento indesejado acontecer. O segundo cenário, chamado aqui de semiativo, é aquele onde as imagens de diferentes câmeras são enviadas para um local centralizado onde são analisadas constantemente pelo pessoal de segurança. Os sistemas semiativos são muito utilizados em prédios e locais onde geralmente é necessária uma ação rápida caso aconteça alguma situação de interesse. Como exemplos de sistemas semiativos a Figura 1 apresenta o centro de monitoramento de contraterrorismo da cidade de Nova York e na Figura 2 apresenta-se um exemplo de um típico sistema de segurança semiativo.

Figura 1 – *Centro de monitoramento de contraterrorismo da cidade de Nova Iorque (Getty Images)*

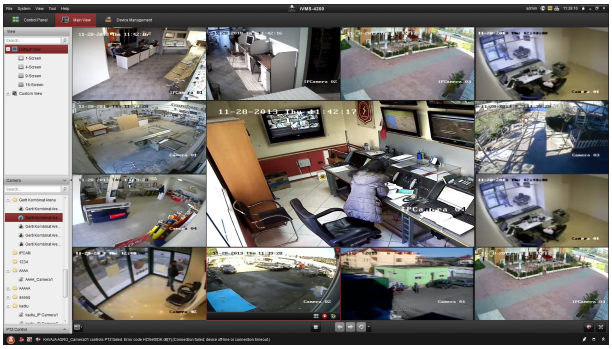


Figura 2 – *Screenshoot de um programa comercial de vigilância (Rogat Security Group, 2011)*

Uma característica comum nos dois cenários anteriormente descritos é o fato das imagens serem analisadas inteiramente por pessoas. No caso de sistemas passivos essa análise é geralmente realizada muito depois do evento de interesse e visa identificar o agressor. Já no caso de sistemas semiativos a análise visa identificar em tempo real o evento de interesse para responder rapidamente. A dificuldade aparece quando o número de câmeras aumenta e o pessoal de segurança deve analisá-las simultaneamente de forma eficiente e confiável.

Atualmente a maioria dos sistemas de segurança por vídeo utilizam várias câmeras conectadas a um computador, o qual organiza as imagens através de um *software*, e as apresenta ao usuário. Uma vez que ditas imagens passam por um computador, é possível

pensar em desenvolver programas informáticos que as analisem e detectem automaticamente eventos de interesse.

Contudo, construir algoritmos capazes de analisar e inferir informações úteis a partir de imagens digitais resultou ser uma tarefa extremamente desafiadora, e tem sido o grande foco de pesquisas no campo de visão computacional e processamento digital de imagens nos últimos anos. Uma das áreas onde esses algoritmos apresentariam grande utilidade é na vigilância assistida por computador, já que permitiriam ao pessoal de segurança se focar em outras atividades e serem alertados unicamente quando o algoritmo detecte alguma situação de interesse.

Vários governos e empresas privadas perceberam a necessidade de desenvolver esses algoritmos e iniciaram projetos de pesquisa para tais fins. Assim, a agência dos Estados Unidos DARPA (do inglês *Defense Advanced Research Projection Agency*) financiou no ano 1999 o projeto VSAM (do inglês *Visual Surveillance And Monitoring project*) (COLLINS *et al.*, 2000) que visava desenvolver tecnologias que fossem aplicáveis em sistemas de segurança por vídeo; no ano 2008, a mesma agência financiou o projeto VIRAT (do inglês *Video and Image Retrieval Analysis Tool*) o qual visava desenvolver uma plataforma de vigilância aérea para detectar automaticamente situações de interesse em cenários de guerra. Agências e centros acadêmicos europeus também iniciaram projetos similares como o ISCAPS (do inglês *Integrated Surveillance of Crowded Areas for Public Security*) no ano 2005, e o projeto ADVISE (do inglês *Advanced Video Surveillance*) os quais tinham como objetivo desenvolver tecnologias para a detecção de situações perigosas em vários tipos de cenários. Finalmente, na França, a rede de transporte público de Paris lançou o CREDS (do inglês *Challenge for Real-time Event Detection Solutions*) visando a criação de algoritmos para a detecção automática de situações de perigo nos trens da cidade, como objetos lançados nos trilhos ou pessoas presas na porta quando o trem está em movimento.

Muitos desses projetos desenvolveram algoritmos que detectam movimento em lugares não permitidos, reconheciam e faziam seguimento de pessoas e objetos como carros, *vans*, etc. A Figura 3 apresenta exemplos dos resultados do projeto VSAM onde são detectados e classificados objetos com movimento. Entretanto, esses algoritmos apresentam uma aplicabilidade muito limitada e a comunidade científica continua trabalhando para desenvolver algoritmos mais rápido, precisos e com aplicabilidade a um número maior de situações.

A detecção e reconhecimento de atividades consideradas como anormais têm chamado muito a atenção da comunidade científica nos últimos anos. A Figura 4 apresenta o número de resultados obtidos no motor de busca da Google Scholar para o termo "*Abnormal Human Behavior Detection and Recognition in video*" em diferentes anos.

O recente aumento de trabalhos relacionados com detecção de comportamento

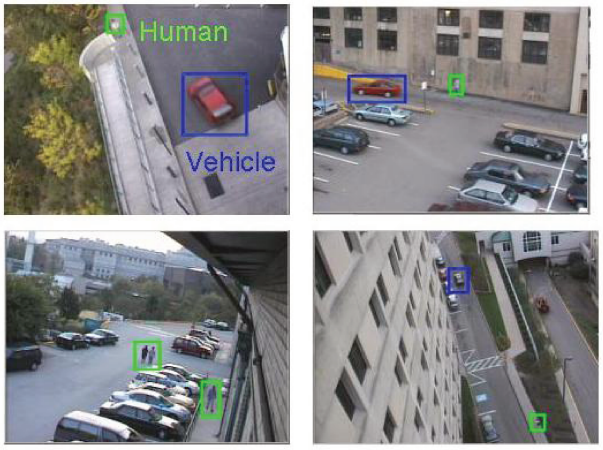


Figura 3 – *Resultados do projeto VSAM.*

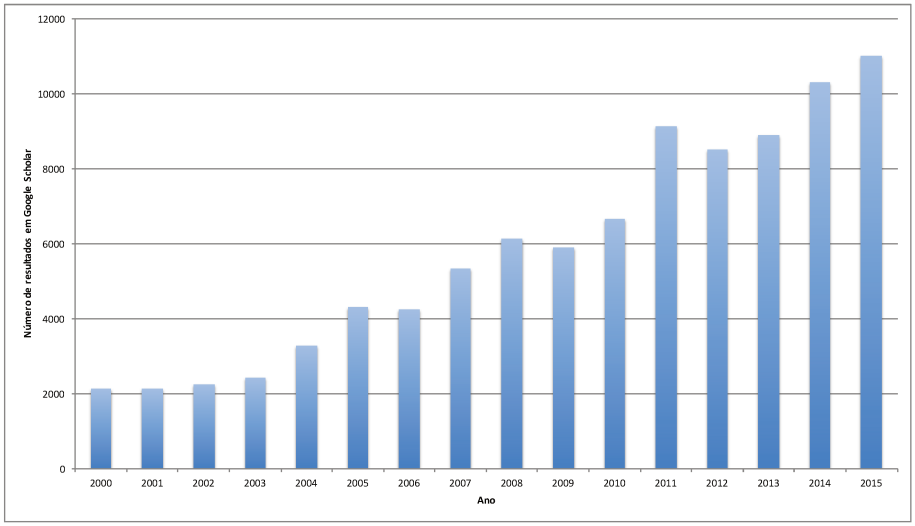


Figura 4 – *Número de artigos encontrados pelo motor de busca de Google Scholar para o termo "Abnormal Human Behavior Detection and Recognition in video" no período de 2000 até Julho de 2016.*

humano anormal pode-se atribuir aos avanços em técnicas de classificação e de aprendizado de máquinas, os quais possibilitaram o desenvolvimento de algoritmos com maior flexibilidade e capacidade de generalização para seu uso em vários tipos de cenários e condições. Porém, atualmente algumas das situações mais desafiadoras para se analisar são as que envolvem grupos de pessoas, onde fenômenos como a oclusão e dinâmicas constantemente sob mudança impossibilitam analisar cada pessoa individualmente e é neste tipo de situações nas quais este trabalho será focado.

1.1 Motivação e considerações inicias

Desenvolver sistemas de vigilância assistidos por computador é uma necessidade da sociedade atual e futura uma vez que a análise de vídeos em busca de situações anormais

usando os sistemas passivos atuais apresentam as seguintes desvantagens:

- A concentração da pessoa que analisa os vídeos tem que dividir-se em muitas imagens ao mesmo tempo e, na maioria das vezes, essa mesma pessoa tem que realizar outras tarefas, como por exemplo responder perguntas, cadastrar visitantes, atender o telefone, etc.
- Situações de interesse acontecem raramente e, portanto, uma pessoa analisando as imagens pode facilmente se distrair perdendo algum evento importante.
- Quanto maior o número de câmeras, maior o número de pessoas necessárias para analisar todas as informações e, portanto, maior o custo operativo do sistema.

As dificuldades anteriormente mencionadas motivam o desenvolvimento de algoritmos que permitam uma detecção automática de situações de anormalidade auxiliando as decisões tomadas pelo pessoal de segurança. No entanto, o primeiro passo para desenvolver esse tipo de algoritmos é definir o quê é uma situação de anormalidade.

O dicionário *online* de português[2] define a palavra anormal como: "*Irregular; contrário ao que é normal. Que foge ou se afasta das normas e dos padrões*". Uma vez que, o que é considerado como comportamento normal, ou dentro dos padrões, depende de uma série de variáveis como locação, tempo, número de pessoas na cena, atividades, etc. o número de possíveis situações anormais detectáveis por um sistema de vigilância pode ser muito alto. Assim, é necessária uma redução dessas possíveis situações de anormalidade, e portanto, neste trabalho serão abordadas quatro situações, as quais, segundo o autor, são consideradas como importantes para qualquer sistema de segurança em ambientes envolvendo multidões. Essas quatro situações serão detalhadas na sequência.

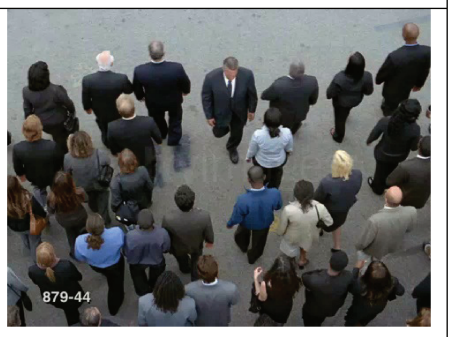
1.1.1 Situações anormais a serem consideradas

A Tabela 1 detalha as quatro situações que visam ser classificadas como anormais pelo sistema proposto neste trabalho.

O primeiro tipo de anormalidade (Caso 1) acontece quando um grupo de pessoas, que inicialmente está caminhando, de repente, começa a correr. Esse tipo de anormalidade é conhecida na literatura como anormalidade global, já que todas as pessoas na cena apresentam o mesmo tipo de comportamento. O segundo tipo de anormalidade acontece quando uma, ou poucas pessoas, apresentam uma velocidade ou direção de deslocamento muito diferente às outras. Neste caso também será detectada como anormalidade quando uma pessoa se desloque em uma região da imagem onde ninguém tinha se deslocado

[2] https://www.dicio.com.br

Tabela 1 – *Anormalidades a serem detectadas.*

Caso 1: Neste cenário é considerado o comportamento das pessoas como sendo normal se todas elas estão caminhando e como anormal se elas aumentarem repentinamente sua velocidade.	
Caso 2: Neste cenário somente uma ou poucas pessoas na cena apresentam comportamento anormal associado à presença, direção ou velocidade de deslocamento.	
Caso 3: Esta situação de anormalidade acontece quando uma pessoa se desloca em direção contrária à direção de deslocamento de uma multidão.	
Caso 4: Esta anormalidade acontece quando um objeto nunca visto ingressa na cena. Este tipo de anormalidade é o único baseado na aparência dos objetos.	

anteriormente. Na literatura esse tipo de anormalidade é chamado de anormalidade local, já que só uma ou poucas pessoas apresentam o comportamento anormal. O terceiro caso de anormalidade apresenta-se quando uma pessoa se desloca em direção contrária à direção de movimentação de uma multidão. Finalmente a quarta anormalidade acontece quando um objeto não visto anteriormente aparece na cena. Esse último tipo de anormalidade depende da aparência e não da movimentação como nos outros três casos.

Cada caso detalhado anteriormente pode-se apresentar com diferentes densidades de pessoas. Por exemplo, o grupo de pessoas na primeira situação pode ser composto unicamente por três indivíduos, enquanto que, em outro cenário, o grupo pode ser composto por milhares de pessoas. Portanto a análise das situações detalhadas nesta seção será restrita às densidades de pessoas apresentadas a seguir.

1.1.2 Densidades a serem consideradas

Os trabalhos desenvolvidos na área de detecção de comportamento anormal em vídeo podem-se classificar em duas categorias: a primeira categoria tenta realizar a detecção em pessoas individuais e a segunda categoria tenta detectar o comportamento anormal em grupos de pessoas. Essa divisão é necessária devido ao fato de alguns algoritmos precisarem segmentar as pessoas para modelar seu comportamento e as técnicas atuais de segmentação falham quando existem muitas oclusões devidas a um alto número de pessoas na cena.

Na categoria de grupo de pessoas é necessária uma segunda divisão já que um grupo de poucas dezenas de pessoas é diferente de um grupo de centenas ou milhares delas. Neste aspecto não existe uma divisão definida nos artigos publicados atualmente, onde o termo multidão (*crowd* em inglês) pode ser aplicado indistintamente a um pequeno grupo de pessoas ou a uma multidão com milhares delas.

Para definir valores quantitativos de densidade de pessoas é possível utilizar o conceito de níveis de serviço definido por (POLUS *et al.*, 1983). A densidade de pessoas em uma cena, isto é, o número de pessoas em uma determinada área, pode ser categorizada em quatro níveis de serviço de acordo com a Tabela 2.

Tabela 2 – *Níveis de serviço, adaptado de (POLUS* et al.*, 1983)*

Descrição do fluxo	Nível de serviço	Densidade (pessoas/m^2)	Velocidade media (metros/seg)
Fluxo livre	A	<= 0.6	1.3 – 1.15
Fluxo restrito	B	0.61 – 0.75	1.15 – 1.1
Fluxo denso	C	0.75 – 2.0	1.1 – 0.8
Fluxo congestionado	D	>= 2.0	0.8 – 0.7

A Figura 5 apresenta exemplos de cenas com diferentes densidades de pessoas e uma possível classificação nos níveis de serviços definidos em (POLUS *et al.*, 1983).

(a) (b)

(c) (d)

Figura 5 – *Exemplos de cenas com diferentes densidades de pessoas. (a) Exemplo de cena com fluxo livre, (b) exemplo de cena com fluxo restrito, (c) exemplo de cena com fluxo denso (d) exemplo de cena com fluxo congestionado.*

O sistema proposto neste trabalho será projetado para detectar anormalidades nas quatro situações detalhadas na seção anterior, com densidade de pessoas nos três primeiros níveis de serviço, ou seja, nas categorias A, B e C, deixando cenas com fluxo congestionado (nível de serviço D) como trabalho futuro.

Definidas as situações e as densidades de pessoas a serem analisadas, na sequência serão apresentados os objetivos deste trabalho.

1.2 Objetivos

Levando em consideração os pontos levantados anteriormente, são definidos os objetivos do trabalho como:

1. Estudar as dificuldades e falências dos algoritmos atualmente disponíveis na literatura em relação à detecção de comportamentos anormais em cenários com multidões.
2. Desenvolver algoritmos focados em superar uma ou varias das falencias encontradas no item anterior em relação à detecção automática das situações de interesse apresentadas na Seção 1.1.1, em cenários com densidade de pessoas nos níveis de serviço A, B ou C, utilizando ferramentas de visão computacional e processamento digital de imagens.
3. Implementar os algoritmos projetados, testando sua performance e comparando os resultados obtidos com aqueles reportados nos trabalhos correlatos em datasets publicamente disponíveis.
4. Projetar e implementar uma arquitetura que permita adicionar novos algoritmos de identificação de situações de anormalidade não contempladas neste trabalho.

1.3 Contribuições do trabalho

Este trabalho visa aportar as seguintes contribuições:

1. Mudança no paradigma de detecção de anormalidades existente atualmente, desenvolvendo algoritmos simples, mas especializados em tipos bem específicos de anomalias.
2. Desenvolvimento de algoritmos baseados em frequências e probabilidades de ocorrência, possibilitando a contínua adesão de novo conhecimento.
3. Proposta de arquitetura por módulos, onde cada módulo se especializará em um determinado tipo de anormalidade, juntando as saídas de cada módulo por meio de um combinador adaptando a saída total do sistema às necessidades específicas do usuário.

No capítulo seguinte serão apresentados os trabalhos mais relevantes disponíveis atualmente na literatura na área de detecção de anomalias em vídeo.

CAPÍTULO 2

Revisão e arquitetura proposta

No artigo (LIPTON *et al.*, 2000) os trabalhos de pesquisa na área de sistemas automáticos de vigilância são classificados em três grupos: detecção e rastreamento, análise de movimento humano e análise de atividade. No mesmo artigo, os autores ressaltam que a análise de atividade se tornará a área mais importante para pesquisa futura em sistemas de vigilância por vídeo, devido a seu grande número de aplicações, entre elas, a detecção de comportamento anormal.

Devido ao grande número de atividades humanas que podem ser registradas em um vídeo, no artigo de (AGGARWALA; RYOO, 2011) é proposta uma divisão em quatro categorias segundo a complexidade da atividade. A primeira categoria é chamada de **gestos**. Os gestos são movimentos elementares das partes do corpo de uma pessoa, por exemplo, levantar o pé do chão, levantar a mão, esticar o braço, etc. A segunda categoria é chamada de **ações**. As ações são atividades realizadas por uma pessoa, e geralmente estão compostas por vários gestos, exemplos de ações são caminhar, correr e pular. A terceira categoria é chamada de **interações**. As interações são ações entre duas ou mais pessoas ou entre uma pessoa e um ou mais objetos. Uma pessoa conversando na frente de outra é um exemplo de interação, da mesma forma que uma pessoa tomando um copo da mesa. Finalmente a quarta categoria é chamada de **atividades em grupos**. Nesta categoria são reunidas as interações entre grupos de pessoas, como por exemplo, várias pessoas correndo ou em uma briga.

Existem vários trabalhos na literatura abordando detecção de anomalias em cada uma dessas categorias, no entanto, o presente trabalho visa detectar anomalias especialmente na última categoria: atividades em grupos; portanto será nessa categoria que a revisão bibliográfica será focada.

Dentro das atividades em grupo pode-se realizar ainda outra divisão em densidades baixas e altas dependendo do número de pessoas que fazem parte da atividade. Quando a densidade de pessoas é baixa, é possível segmentá-las e realizar a análise de comporta-

mento de forma individual. No entanto, quando o número de pessoas na cena aumenta a tarefa de segmentar cada indivíduo torna-se muito mais difícil, demandando novas abordagens. Por conseguinte, a revisão da literatura será dividida em duas secções: detecção de comportamento anormal em cenas com baixa densidade e detecção de comportamento anormal em cenas com média e alta densidade.

Nos diferentes trabalhos de pesquisa o termo "comportamento" (*behavior* em inglês) é usualmente referido a ações observáveis de agentes, como pessoas; e o termo "anormal" (*abnormal* em inglês) são comportamentos que fogem da norma e, portanto, chamam a atenção de um ser humano que observa a cena, geralmente esse tipo de comportamento é facilmente identificável por uma pessoa. Embora a anormalidade seja um conceito ligado diretamente ao contexto da cena em (VARADARAJAN; ODOBEZ, 2009) é definida a anormalidade como *uma ação realizada em um local não usual em um tempo não usual*.

2.1 Reconhecimento de comportamento anormal em cenas com baixa densidade de pessoas

Em (POPOOLA; WANG, 2012) é apresentada uma revisão das técnicas utilizadas para o reconhecimento de comportamento anormal humano em vídeo. No artigo é evidenciado o grande interesse da comunidade científica em desenvolver técnicas e algoritmos de reconhecimento desse tipo de comportamento. Essas técnicas poderiam, no futuro, ajudar a diminuir o tempo de resposta dos operadores frente aos problemas de segurança, monitorando lugares públicos como aeroportos, estações de trem, centros comerciais, instalações militares entre outros; assim como ajudar na vigilância de idosos ou crianças pequenas. Na sequência serão apresentados alguns trabalhos relevantes na área de identificação de comportamento anormal em cenas com densidades de pessoas baixas. Em geral as técnicas aqui apresentadas requerem a segmentação de cada indivíduo para sua análise e, portanto, não podem ser generalizadas para cenas com altas densidades de pessoas.

Um dos primeiros trabalhos em detecção de atividade anormal foi publicado em 2004 (ZHONG *et al.*, 2004). Nesse algoritmo o primeiro passo é a detecção de objetos com movimento através de um filtro espaço-temporal (dois filtros gaussianos) proposto pelos autores. A informação de velocidade assim como a cor e a textura dos objetos segmentados são utilizadas para gerar um modelo de comportamento. O algoritmo proposto tem uma abordagem não supervisionada, uma vez que primeiro são utilizados exemplos contendo só comportamento normal para criar vários modelos os quais serão comparados com novas amostras. Para determinar a semelhança entre uma nova amostra e todos os modelos de comportamento normal gerados na fase de treinamento, é utilizada uma técnica de processamento da linguagem natural chamada LSI (*Latent Semantic Indexing*) (DE-

ERWESTER *et al.*, 1990), essa técnica é empregada para determinar as palavras-chaves em um documento e criar um dicionário que permita classificar outros textos dentro de categorias. De forma similar, no artigo utiliza-se a técnica para estabelecer as ações mais comuns e detectar anomalias quando novas amostras estejam muito distantes dos modelos obtidos. Uma vantagem importante dessa técnica é a utilização de uma abordagem de aprendizado não supervisionado, evitando a criação de modelos estatísticos ou o uso de redes neurais que precisam de muita mais informação. Como desvantagem do método pode-se enunciar a elevada complexidade e a utilização de vários algoritmos que geram um custo computacional alto. O método visa detectar comportamento anormal em vídeos gravados antecipadamente, ou seja, não é utilizado em uma sequência de vídeo em tempo real.

No artigo de (NIU *et al.*, 2004) é apresentada uma técnica para a detecção e reconhecimento de atividades humanas em ambientes externos. No artigo as atividades são classificadas como comportamento normal ou comportamento suspeito. Os autores utilizam uma abordagem estatística baseada no paradigma de hipóteses-verificação usados em técnicas como filtragem do Monte Carlo (KITAGAWA, 1996), filtro de partículas (PITT; SHEPHARD, 1999) e algoritmos genéticos entre outros. A técnica gera um modelo de probabilidade que é atualizado a cada nova informação proveniente dos sensores, neste caso a informação que ingressa no modelo probabilístico é a posição e a velocidade do centroide dos objetos segmentados na cena. Para obter a informação da trajetória e velocidade das pessoas é empregada a diferença simples entre *frames*.

Em (XIANG; GONG, 2005) é apresentado um algoritmo para a modelagem e detecção automática de comportamento anormal utilizando uma abordagem não supervisionada. O algoritmo modela o fundo da cena por meio de Mistura de Gaussianas (STAUFFER; GRIMSON, 1999) e análise de componentes conexos para obter um vetor de 7 características: o centroide do componente conexo, o tamanho do componente conexo, a razão entre o número de pixels do componente conexo e a área do *bounding box* que contém esse componente e finalmente os momentos de primeira ordem do componente conexo. Esses sete parâmetros são utilizados para gerar um conjunto de padrões de comportamento que serão posteriormente utilizados para treinar um modelo MOHMM (*Multi-Observation Hidden Markov Model*) que represente os diferentes tipos de comportamento normal nos vídeos de referência. O método apresentado nesse artigo gera modelos para comportamentos complexos, mas ele precisa segmentar cada pessoa.

No artigo (BOUTTEFROY *et al.*, 2010) é apresentado um algoritmo para a detecção de comportamento anormal em estradas. O comportamento anormal detectado nesse trabalho acontece quando uma pessoa cruza a estrada seguindo um caminho não permitido. Para modelar a direção de movimentação das pessoas os autores utilizam a posição e o vetor de velocidade junto com o MRF (*Markov Random Fields*). Os resultados apresen-

tados no artigo mostram a detecção do comportamento anormal em cenas com densidade de pessoas muito baixas e com visão muito clara da estrada, sem apresentar maior desafio no processamento.

Em (WALHA *et al.*, 2013) é apresentada uma técnica de detecção de comportamento anormal em vídeo adquirido desde um veículo aéreo. Nesse trabalho é usado o SIFT (*Scale Invariant Feature Transform*), que é um algoritmo de detecção e descrição de pontos de interesse que são invariantes a mudanças de escala e rotação (LOWE, 2004), para estabelecer pontos correlatos entre um *frame* e o seguinte, dessa maneira é obtida uma estimativa do movimento do veículo aéreo assim como o movimento das pessoas na cena. Devido ao movimento da câmera, objetos fixos na cena apresentam deslocamento entre *frames*, para determinar quais objetos são fixos e quais apresentam movimento os autores assumem que os valores de deslocamento de objetos com movimento são muito maiores do que os valores de deslocamento dos objetos fixos. Os autores utilizam o filtro de Kalman para realizar o seguimento dos objetos com movimento na cena. Para detectar comportamento anormal os autores utilizam cinco informações: o identificador de movimento do objeto, o tempo no qual o objeto começou a se deslocar, o tempo no qual o objeto parou de se deslocar, o tempo total de movimentação e o tipo de atividade do objeto. Essas cinco informações são comparadas com uma base de dados obtida antecipadamente e é determinada uma anomalia na cena. O trabalho não apresenta resultados de detecção de anomalias e o algoritmo precisa segmentar cada pessoa para modelar seu movimento.

No artigo (WANG *et al.*, 2014) é apresentado um algoritmo para a análise de trajetórias e identificação de comportamento anormal. O deslocamento das pessoas é modelado para, posteriormente, detectar o comportamento anormal. O primeiro passo do algoritmo é segmentar os objetos que não pertencem ao fundo, isso é feito utilizando o algoritmo de Mistura de Gaussianas (STAUFFER; GRIMSON, 1999). No segundo passo é usada a análise de componentes conexos para determinar a trajetória do centroide das pessoas na cena, a distância entre os centroides do mesmo componente conexo em dois *frames* consecutivos é calculada e armazenada em um vetor. A seguir, a informação do vetor de distâncias é codificada e comprimida por meio de um algoritmo de codificação chamado SAX (*Symbolic Aggregate Approximation*). O SAX é um algoritmo que gera representações simbólicas eficientes ao converter dados de séries temporais em uma cadeia de caracteres dentro de um alfabeto finito, de acordo com uma série de regras bem definidas (LIN *et al.*, 2003). Assim, as distâncias entre centroides de um componente conexo em uma sequência de *frames* são reduzidas e codificadas em uma série de letras, ocupando muita menos memória. No passo seguinte a codificação é utilizada para treinar um modelo de Markov de primeira ordem, onde são utilizadas cenas com comportamento normal para obter a probabilidade de, no estado atual, apresentar uma determinada letra tendo em conta a letra apresentada no estado anterior. Para determinar quando aparece comportamento anormal é calculada a probabilidade de uma sequência de letras serem iguais à mesma

sequência de letras obtidas no conjunto de treinamento. O método apresentado nesse artigo é simples e computacionalmente eficiente, mas só leva em conta a trajetória e seu uso está limitado a cenas de baixa densidade. Outra desvantagem do método é que, de acordo com os autores, os parâmetros do SAX influenciam de forma importante o desempenho do algoritmo e os valores corretos destes parâmetros são difíceis de obter.

Em (CHEN *et al.*, 2015) é apresentada uma técnica onde a posição e a velocidade de pontos de interesse no *foreground* são utilizados para calcular métricas de ângulos e distâncias entre ditos pontos. Um modelo de movimento de partículas é criado e a detecção de comportamento anormal é realizada utilizando um limiar.

Na seção a seguir serão apresentados os trabalhos que visam tratar cenas com densidades maiores de pessoas.

2.2 Reconhecimento de comportamento anormal em cenas com média e alta densidade de pessoas

Um dos primeiros trabalhos a tratar o problema de detecção de comportamento anormal em grupo foi (ANDRADE *et al.*, 2006). A abordagem utilizada pelos autores é baseada no modelo do comportamento normal e posterior detecção, por meio de um *threshold*, de comportamento anormal. A primeira etapa da técnica é modelar o fundo e obter os elementos do *foreground* (elementos que não são parte do fundo da cena) por meio do algoritmo de Mistura de Gaussianas (STAUFFER; GRIMSON, 1999), a imagem obtida com o *foreground* é utilizada como máscara para filtrar os vetores de interesse após ser calculado o fluxo óptico na cena. Na segunda etapa é utilizada a Análise de Componentes Principais (PCA pelas siglas em inglês) (JOLLIFFE, 2002), a qual é uma técnica de redução de dimensionalidade utilizada para comprimir dados. Os autores utilizam o PCA para garantir sempre o mesmo número de vetores de fluxo óptico que serão utilizados na máquina de classificação. Para a classificação de comportamento normal ou anormal os autores treinam um MOHMM (*Multiple Observation Hidden Markov Model*) (XIANG; GONG, 2005). Assim é calculado um valor de *threshold* com a probabilidade de uma sequência de *frames* em um vídeo seja parte do modelo normal definido pelo MOHMM. O método apresentado nesse artigo tem uma abordagem não supervisionada; como desvantagem pode-se mencionar a necessidade de treinar vários modelos probabilísticos para as diferentes classes de comportamento anormal obtidas pelo MOHMM tornando o método mais complexo e computacionalmente custoso; além disso o método é muito sensível ao número de vetores resultantes após a compressão com o PCA.

Em (MEHRAN *et al.*, 2009) é apresentada uma técnica de detecção de comportamento anormal em cenas com multidões. A técnica é chamada de modelo por forças

sociais (*Social Force Model*). Nesta técnica é calculado o fluxo óptico entre *frames*, a seguir é utilizado um algoritmo que cria uma grade de partículas cuja velocidade e direção dependem dos valores do fluxo óptico em uma janela escolhida. A movimentação dessas partículas cria uma interação entre elas a qual é modelada como uma força de fluxo. Essa força de fluxo é mapeada de novo na imagem e são criadas palavras visuais (cubos de tamanho $n \times n \times T$ onde $n \times n$ é uma região de interesse dentro da imagem e T um determinado número de *frames*). Na sequência essas palavras visuais são utilizadas para criar uma sacola de palavras (*Bag of Words*), e junto com o algoritmo LDA (*Latent Dirichlet Allocation*) (BLEI *et al.*, 2003) é obtida a máquina de classificação de comportamento anormal. O método descrito nesse artigo apresenta melhores resultados na identificação de comportamento anormal comparado com os trabalhos que utilizam somente fluxo óptico, entretanto a técnica requer um grande número de parâmetros ajustáveis e esses parâmetros mudam dependendo da cena. Também utiliza modelos probabilísticos complexos que limitam sua aplicação em sistemas de tempo real.

Em (MAHADEVAN *et al.*, 2010) é apresentada uma técnica de detecção de comportamento anormal em multidões baseada no modelo de texturas na cena. A metodologia adotada nesse trabalho é obter um modelo na etapa de treinamento e calcular a probabilidade de novas amostras fazer parte dele. Se a nova amostra possui baixa probabilidade de pertencer ao modelo, então é classificada como anomalia. Nesse trabalho o modelo gerado é composto pela mistura de texturas dinâmicas (MTDs) (DORETTO *et al.*, 2003). As texturas dinâmicas são modelos probabilísticos conjuntos compostos de aparência e dinâmica das texturas dos objetos da cena. Os resultados apresentados no artigo identificam os objetos anormais nas cenas, entretanto a sua localização não é precisa e pode levar facilmente a falsos positivos, além disso, os modelos estatísticos são computacionalmente custosos e o modelo treinado não é atualizado para levar em conta novos comportamentos.

Em (SU *et al.*, 2012) é apresentada uma técnica onde a cena é modelada utilizando a mesma ideia de (MEHRAN *et al.*, 2009), uma grade de partículas é utilizada para modelar as interações entre as partículas e dessa forma determinar o comportamento anormal. Os autores desse artigo desenvolvem o modelo matemático tomando as partículas como parte de um fluido, assim são definidos um campo de fluido espaço-temporal e um campo de forças espaço-temporal. Como máquina de classificação é utilizado o algoritmo LDA (BLEI *et al.*, 2003) para discriminar o comportamento anormal. A técnica apresentada nesse artigo está baseado em modelos matemáticos complexos que aumentam o custo computacional.

No artigo (WANG; SNOUSSI, 2012) é apresentada uma técnica para a detecção de mudanças nos padrões de movimentação de multidões. A técnica utiliza o fluxo óptico para criar, o que os autores chamam de Histograma da Orientação do Fluxo Óptico (HOFs pelas siglas em inglês). Na primeira etapa do algoritmo as direções dos vetores

do fluxo óptico são quantizadas em intervalos fixos e organizadas em um histograma, o histograma é utilizado para treinar uma SVM (*Support Vector Machine*) de uma única classe e gerar o modelo da direção de deslocamento das pessoas na cena. Na segunda parte do algoritmo os histogramas gerados por novos *frames* são classificados pela SVM como normais ou anormais. A técnica é simples de implementar e não precisa de modelos estatísticos complicados, porém, a técnica só leva em conta a direção de movimentação, desconsiderando a magnitude, portanto limitando suas aplicações.

No artigo de (REN *et al.*, 2012) é apresentada uma abordagem utilizando conceitos de entropia, energia e estados para detectar comportamento anormal e sua localização na cena. O algoritmo utiliza fluxo óptico para definir a certeza de comportamento e a entropia de comportamento. A análise de comportamento normal é baseada na probabilidade de uma partícula mudar de estado de acordo a sua entropia atual. No artigo a certeza de comportamento é utilizada para determinar o momento no qual aparece uma anormalidade e a entropia de comportamento é utilizada para determinar sua localização na cena. Uma desvantagem desse método é que os parâmetros dependem da cena, portanto devem ser ajustados quando existir alguma mudança nas condições nas quais foi treinado o modelo.

Em (ZHANG *et al.*, 2013) é apresentada uma técnica para a detecção de 4 tipos de anormalidade definidas pelos autores: debandada, queda, briga e pânico. Para a identificação de cada tipo de anormalidade é utilizada uma rede neural do tipo MLP (*Multilayer Perceptron*) com múltiplas saídas. No primeiro passo do algoritmo são detectados os objetos com movimento na cena através de uma técnica própria de detecção de movimento. É calculado o fluxo óptico para determinar a direção e a velocidade de deslocamento das pessoas. A rede MLP é treinada a partir de informações espaço-temporais do fluxo óptico em uma região de tamanho $n \times n$ na imagem durante um número T de *frames*. A técnica apresentada nesse artigo tem como novidade a classificação do comportamento anormal em quatro categorias, no entanto apresenta como desvantagem a necessidade de dados rotulados para a etapa de treinamento da rede, o que significa que são necessárias amostras da mesma cena com as diferentes categorias de comportamento anormal, as quais, geralmente, não estão disponíveis.

No artigo (XU *et al.*, 2014) é proposta uma técnica de detecção de comportamento anormal em cenas com multidões. A abordagem adotada pelos autores do artigo é detectar as anomalias baseando-se em um *framework* para modelar atividades-padrão hierárquico. O modelo leva em conta a informação espacial e temporal. É utilizada uma abordagem não supervisionada onde não são necessárias amostras rotuladas, entretanto é necessária uma etapa de treinamento *off-line* onde todas as amostras pertencem a exemplos de comportamento normal. São utilizados histogramas do fluxo óptico para criar uma estrutura de níveis que geram uma função unificada de energia utilizada como máquina de classificação.

O artigo (YANG *et al.*, 2014) apresenta uma técnica para detectar comportamento anormal de dois tipos: dispersão e distúrbios. Na técnica é utilizado o fluxo óptico junto com o algoritmo de extração de fundo por Mistura de Gaussianas. Os autores utilizam conceitos de entropia e energia do fluxo para calcular dois parâmetros, cada parâmetro modela um tipo de anormalidade detectável pela técnica. Uma das principais desvantagens desse método é a necessidade de calibrar de novo todo o modelo se a câmera é movimentada levemente, já que são utilizadas medidas de perspectiva e parâmetros de posição da câmera no modelo. O algoritmo também utiliza muitos parâmetros, dois deles são selecionados *a priori* o que leva uma sequência de tentativa e erros até o algoritmo funcionar corretamente.

Em (FRADI; DUGELAY, 2014) é apresentada uma técnica para a detecção de comportamento anormal em multidões onde é utilizado o algoritmo FAST (*Features from Accelerated Segment Test*) para a obtenção de pixels que são muito diferentes aos pixels da vizinhança. O FAST é utilizado para a extração de pontos de interesse ou *features*. A seguir é utilizado o algoritmo RLOF (*Robust Local Optical Flow*) para obter o fluxo óptico dos pixels *features*. A informação obtida do fluxo óptico é utilizada para calcular as trajetórias desses pixels. No artigo os autores utilizam a densidade a velocidade e a orientação dos pixels *features*. A densidade dos pixels *features* é utilizada para estimar a densidades de pessoas na multidão, quando maior densidade de pixels *features* maior número de pessoas na multidão. A densidade, a velocidade e a direção dos pixels *features* são quantizadas para criar histogramas unidimensionais. Os autores utilizam a categorização de densidade feita por (POLUS *et al.*, 1983), portanto o histograma da densidade tem 5 intervalos, um para cada nível de serviço (Tabela 2). As velocidades estão quantizadas em 5 classes: muito lento, caminhando, caminhando rápido, correndo e correndo muito rápido. A direção é quantizada em intervalos de 45 graus, gerando um histograma de 8 barras. Os três histogramas são gerados a cada novo *frame*. Para detectar as mudanças na densidade, velocidade ou direção de movimento são armazenados histogramas correspondentes aos histogramas gerados nos anteriores K *frames*. Assim o histograma de densidade obtido no último *frame* é comparado com os K histogramas de densidade anteriores; o último histograma de velocidade é comparado com os K histogramas de velocidade anteriores e o último histograma de direção é comparado com os K histogramas de direção anteriores. As mudanças nos histogramas é detectada através de uma métrica de similaridade descrita em (ALMEIDA; JUNG, 2013). Se a medida de similaridade entre o novo *frame* e os anteriores é muito baixa então é detectada uma anomalia. O algoritmo tem a limitante de se basear unicamente na informação dos últimos *frames* e não leva em conta um maior histórico. O sistema proposto nesse artigo também precisa dados rotulados para treinar uma SVM os quais, geralmente, não estão disponíveis.

A Tabela 3 resume as diferentes técnicas detalhadas nesta revisão.

Tabela 3 – *Resumo dos trabalhos correlatos.*

Autor	*Features*	Método de classificação	Treinamento não supervisionado	Treinamento supervisionado	Informação de movimento	Informação de aparência
mehran2009	Fluxo óptico. Modelo de interações entre partículas.	*Bag of words. Latent Dirichlet Allocation.*	x		x	
mahadevan2010	*Dynamic Textures* (DT)	*Bag of words. Latent Dirichlet Allocation.*	x			x
bouttefroy2010	Posição e trajetórias de pontos de interesse (*keypoints*)	Probabilístico: *Gaussian Markov random field mixture.*	x		x	
ren2012	Fluxo óptico	*Threshold* estático.	x		x	
su2012	Modelo de energias entre partículas.	*Bag of words. Latent Dirichlet Allocation.*	x		x	
wang2012	Fluxo óptico.	SVM de uma classe.	x		x	
zhang2013	Campo de fluxo espaço temporal.	Rede Neural (MLP).		x	x	

Tabela 3 – *Resumo dos trabalhos correlatos (continuação)*

Autor	*Features*	Método de classificação	Treinamento não supervisionado	Treinamento supervisionado	Informação de movimento	Informação de aparência
alvar2014	Trajetórias	*Clustering (Dominant Sets)*	x		x	
liu2014	Dynamic Textures (DT). Pontos de interesse (Keypoints)	*Sparce coding*	x		x	
xu2014	Histogramas da orientação do fluxo óptico. (HOOF). Salient Activities.	*Threshold* estático	x		x	
li2014	Mixture of dynamics textures. (MDT)	Modelo probabilístico com *threshold* estático				x
chen2015	Fluxo óptico. Trajetórias. (Dense trajectories)	Fuzzy C-Means	x		x	
kaltsa2015	Fluxo óptico. Histograms of Oriented Swarms (HOS). Histograms of Oriented Gradients (HOG).	*Support Vector Data Description* (SVDD)	x		x	x
choi2016	Histograms of Oriented Gradients (HOG) Histograms of Optical Flows (HOF)	*Deep autoencoder* com *Threshold* fixo.		x	x	x

2.3 Discussão

A partir da Tabela 3 é possível verificar que o fluxo óptico é o tipo de *feature* mais utilizado nos trabalhos. Algumas propostas encontradas na literatura como (MEHRAN *et al.*, 2009), (KIM; GRAUMAN, 2009), (RYAN *et al.*, 2011) e (NASCIMENTO *et al.*, 2013) utilizam unicamente o fluxo óptico para obter informações de movimento como a velocidade e/ou direção.

Outras abordagens como as presentes em (MAHADEVAN *et al.*, 2010), (ROSHTKHARI; LEVINE, 2013) e (LI *et al.*, 2014), utilizam unicamente informações de aparência e suas dinâmicas através do tempo. Esses algoritmos utilizam (HOG, *histograms of oriented gradientes*) (DALAL; TRIGGS, 2005) ou texturas dinâmicas (DT) (DORETTO *et al.*, 2003) como *features*.

Outras abordagens mais recentes como (ITO *et al.*, 2012), (KALTSA *et al.*, 2015) e (HASAN *et al.*, 2016) levam em conta tanto as informações de movimento quanto às de aparência; nesses casos Histogramas de Orientações do Fluxo Óptico (HOOF, *Histogram of Oriented Optical Flow*) (CHAUDHRY *et al.*, 2009) são utilizados para codificar as informações de movimento, enquanto os HOG são utilizados para gerar modelos de aparência.

Comparando os resultados obtidos nas abordagens descritas anteriormente pode-se concluir que os algoritmos que levam em conta tanto as informações de movimento quanto as de aparência geralmente apresentam um melhor desempenho (KALTSA *et al.*, 2015). Isto em conformidade com a máxima de que quanto mais informação se possui sobre um tema, maior probabilidade de tomar a decisão correta. No entanto, maior informação implica modelos mais complexos e vetores de *features* maiores gerando um custo computacional muito alto.

Também, a partir da Tabela 3, é possível atestar a preferência da comunidade em utilizar algoritmos não supervisionados. Isto é devido ao fato de muitos dos trabalhos abordarem a detecção de anormalidades como um problema de detecção de amostras fora do padrão (*outliers*).

Os algoritmos não supervisionados permitem gerar um modelo de "normalidade" a partir de amostras unicamente com situações consideradas como normais; em contraste, os algoritmos de classificação supervisionados requerem amostras conhecidas tanto de situações normais quanto de anormais, as quais, em geral, não são possíveis de obter na fase de treinamento devido a que comportamentos anormais são raros e, portanto, as amostras deste tipo de comportamento são limitadas ou às vezes nulas.

Da revisão da literatura foram identificadas duas limitantes presentes nas abordagens atuais.

A primeira limitante tem a ver com a inclusão de novo conhecimento no modelo. As abordagens que apresentam as melhores taxas de classificação e localização de anormalidades são aquelas baseadas em técnicas de aprendizado de máquinas não supervisionado. Nessas técnicas o conceito de *anormal* é definido como aquele comportamento que **nunca** foi visto nas amostras de treinamento. Isso implica que na etapa de treinamento devem acontecer, pelo menos uma vez, todos os possíveis comportamentos normais. Eis porque são necessárias todas as amostras de comportamento normal para que, no caso de redes neurais, sejam obtidos os parâmetros da rede, e, no caso da SVM, sejam obtidos os vetores suporte. A desvantagem dessa metodologia é que tanto os parâmetros da rede neural quanto os vetores suportes da SVM são dificilmente atualizáveis. Atualizar o classificador requer treiná-lo de novo usando as novas amostras junto com as anteriores. Consequentemente, todas as amostras, tanto novas como passadas, devem ser armazenadas, e o classificador treinado de novo a cada adesão de informação.

A segunda limitante aparece porque abordagens atuais visam detectar as anormalidades sem se importar da sua natureza. Isto é, em uma única cena podem existir diferentes tipos de comportamentos anormais, por exemplo: uma pessoa correndo enquanto outra caminha em uma região não permitida. As abordagens não supervisionadas atuais detectariam as duas anormalidades, no entanto não identificam sua natureza. A classificação do tipo de anormalidade poderia fornecer informações mais precisas ao usuário que poderia tomar decisões diferentes dependendo dos tipos de anormalidades presentes na cena.

O presente trabalho visa superar essas duas limitantes anteriormente apresentadas implementando algoritmos que produzam modelos facilmente atualizáveis e classificadores que possam discriminar o tipo de anormalidade.

Para superar a primeira limitante serão implementados algoritmos baseados em frequências ou probabilidades de ocorrência, assim, os modelos serão atualizados constantemente. Com essa abordagem é possível ampliar o conceito de anormal e não só identificar anormalidades que nunca aconteceram anteriormente, mas também classificar como anormais aqueles comportamentos que são muito pouco frequentes ou improváveis de acontecer.

Para superar a segunda limitante se implementarão algoritmos simples, mas especializados em um tipo específico de anormalidade, assim, por exemplo, será implementado um algoritmo para detectar mudanças na direção de deslocamento e outro algoritmo diferente para identificar anormalidades na velocidade das pessoas, permitindo informar o tipo de anormalidade.

Visando desenvolver um sistema onde esses algoritmos funcionem de forma independente, mas suas saídas possam ser unificadas e customizadas de acordo com certos critérios, é proposta uma arquitetura baseada em módulos detalhada na sequência.

2.4 Arquitetura proposta

O sistema de detecção de anormalidades proposto neste trabalho será implementado utilizando uma arquitetura por módulos, onde cada módulo especializar-se-á em um determinado tipo de anormalidade. A saída de cada módulo será analisada por um combinador que as juntará através de algum algoritmo que permita customizar a saída final do sistema segundo as necessidades do usuário.

A Figura 6 apresenta um esquema da arquitetura proposta, onde a entrada é pré-processada para obter as informações de velocidade, direção e aparência dos objetos na cena, ditas informações são passadas aos diferentes módulos que as analisarão e determinarão se existe anormalidade ou não. A saída de cada módulo será analisada por um combinador que obterá a saída final do sistema.

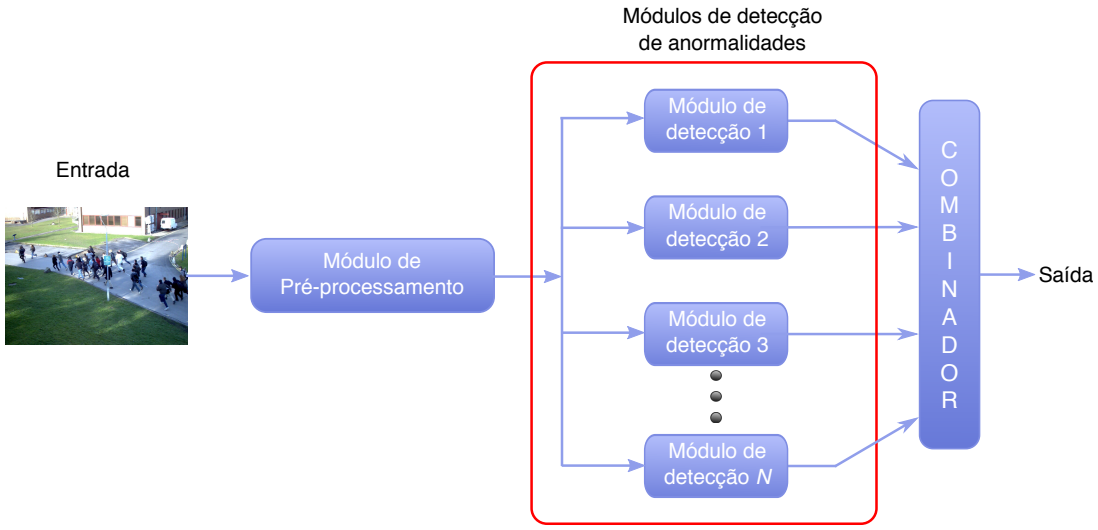


Figura 6 – *Arquitetura proposta para a detecção de anormalidades. Cada módulo especializar-se-á em um tipo específico de anormalidade, a saída de cada módulo é juntada por um combinador o qual obterá a saída definitiva do sistema.*

A arquitetura baseada em módulos apresenta as seguintes vantagens:

1. Possibilidade de executar módulos que compartilham a mesma entrada de forma paralela. Diminuindo o tempo de processamento de cada *frame*.
2. Possibilidade de incrementar facilmente algoritmos de detecção de anomalias não contemplados neste trabalho.
3. Possibilidade de atualizar os algoritmos existentes sem necessidade de modificar os restantes.

Como exemplo da primeira vantagem, percebe-se que as três primeiras situações de anormalidade a serem detectadas neste trabalho, descritas na Seção 1.1.1, são baseadas

em informações de velocidade e direção de movimento o que permitiria processar a mesma informação de entrada nos três módulos de forma paralela.

No capítulo seguinte será detalhado cada módulo do sistema assim como os algoritmos desenvolvidos para detectar cada tipo de anormalidade.

CAPÍTULO 3

Algoritmos de detecção de anormalidades

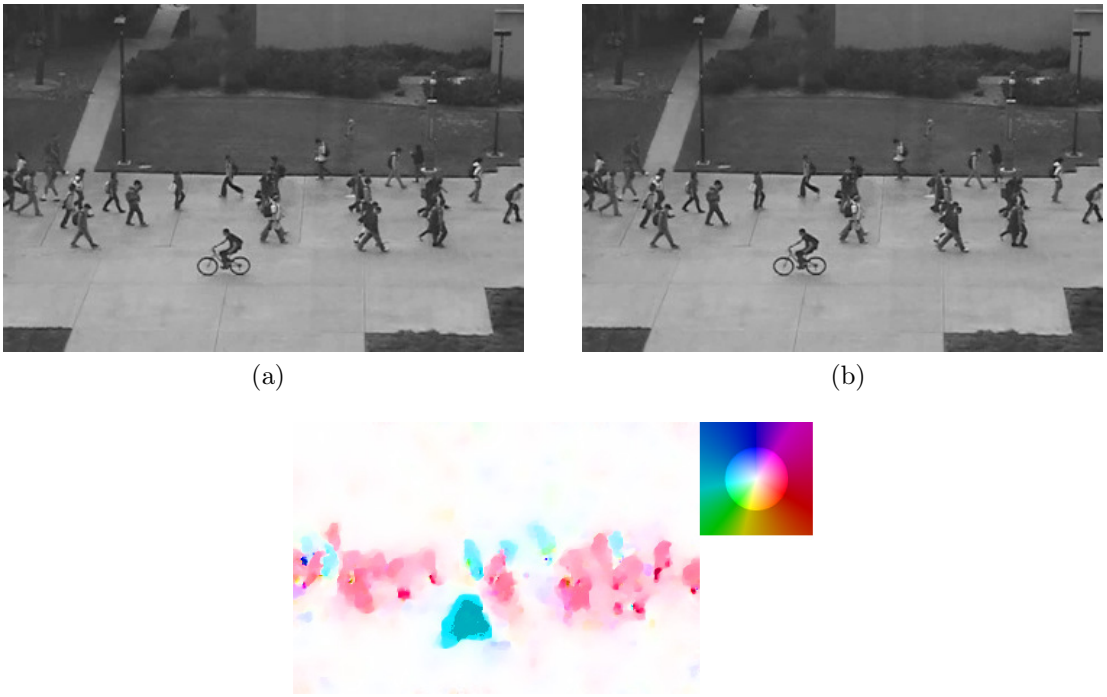
Neste capítulo serão descritos os algoritmos desenvolvidos para a detecção de anormalidades assim como o pré-processamento realizado à entrada para obter as informações de velocidade, direção e aparência dos objetos da cena.

3.1 Pré-processamento

Os primeiros três tipos de anormalidade são baseados nas informações de velocidade e direção de movimento das pessoas na cena e, como pode-se concluir da revisão bibliográfica apresentada no Capítulo 2, o algoritmo mais utilizado na análise de movimentação de objetos em vídeo é conhecido como fluxo óptico (*optical flow* em inglês).

O fluxo óptico é uma aproximação do campo vetorial de velocidades (conhecido como *Motion Field*) dos pixels em uma sequência de imagens gerado pelo movimento relativo entre a câmera e os elementos da cena observada. Em outras palavras, o fluxo óptico determina quanto um pixel movimentou-se entre dois *frames* consecutivos. É possível obter o fluxo óptico para imagens 2D e para volumes 3D, no entanto neste trabalho só será levado em conta o caso 2D. No Anexo B serão apresentados os diferentes algoritmos para obtenção do fluxo óptico e suas limitações. Os algoritmos de fluxo óptico são geralmente divididos em dois grupos: aqueles que obtém o fluxo óptico para cada pixel (fluxo óptico denso) e aqueles que obtém o fluxo óptico para um determinado número de pixels (fluxo óptico esparso). Neste trabalho o fluxo óptico é obtido utilizando o algoritmo de Brox (BROX *et al.*, 2004), o qual calcula o fluxo óptico denso, ou seja, obtém um vetor de movimentação para cada pixel da imagem de entrada. Na Figura 7 apresenta-se um exemplo do fluxo óptico obtido a partir de dois *frames* consecutivos.

É possível que a cena analisada contenha objetos com movimento repetitivo e de pouco interesse, como galhos de árvores ou bandeiras em movimento. Esses tipos de elementos geram informações de fluxo óptico que seriam processadas pelos módulos constan-



(a) (b)

(c)

Figura 7 – *Exemplo de obtenção do fluxo óptico. (a) frame no instante* $t-1$*, (b) frame no instante* t*, (c) fluxo óptico usando o código de cores Middlebury. A direção do vetor de movimento é codificada com uma cor (hue) e sua magnitude pela intensidade (saturation).*

temente, desperdiçando recursos computacionais e gerando possíveis erros de classificação. Além do anterior, o ruído intrínseco acrescentado no processo de aquisição da imagem pode gerar valores de fluxo óptico em pixels que não apresentam nenhum movimento. Com o fim de reduzir o ruído e o número de pixels a serem processados pelos módulos diminuindo a carga computacional, é utilizado um algoritmo chamado de subtração de fundo.

Os algoritmos de subtração de fundo (*background subtraction* em inglês) visam segmentar elementos fixos na cena como prédios, carros estacionados, árvores, etc. daqueles que são temporais como pessoas caminhando ou carros andando. Os elementos fixos são chamados de elementos *background* e os elementos temporais são chamados de elementos *foreground*. Existem vários algoritmos que obtém os elementos *foreground* de uma cena e uma descrição mais detalhada deles pode-se encontrar no Anexo A. Os algoritmos de subtração de fundo geram uma máscara que pode ser utilizada para reduzir o número de vetores do fluxo óptico que serão processados pelos módulos.

A Figura 8 detalha a obtenção do fluxo óptico reduzido a partir do fluxo óptico denso e a máscara do *foreground*.

Neste trabalho os algoritmos de fluxo óptico e segmentação de fundo recebem como entrada imagens em tons de cinza, pelo qual, se a imagem de entrada está em formato de cores RGB, ela é convertida a tons de cinza. Também é realizada uma redução do ruído

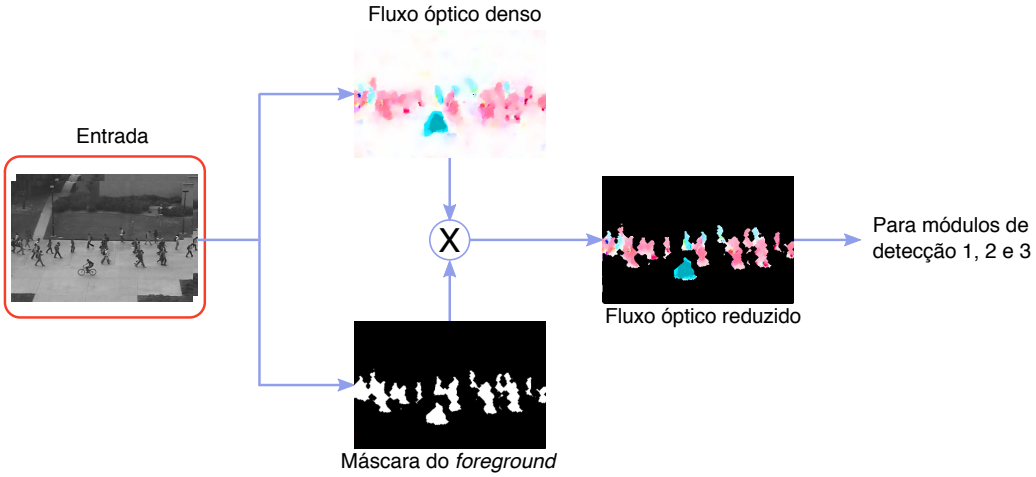


Figura 8 – *Obtenção do fluxo óptico reduzido.*

na imagem em cinza usando um filtro gaussiano.

A Figura 9 apresenta a arquitetura do sistema detalhando o módulo de pré-processamento

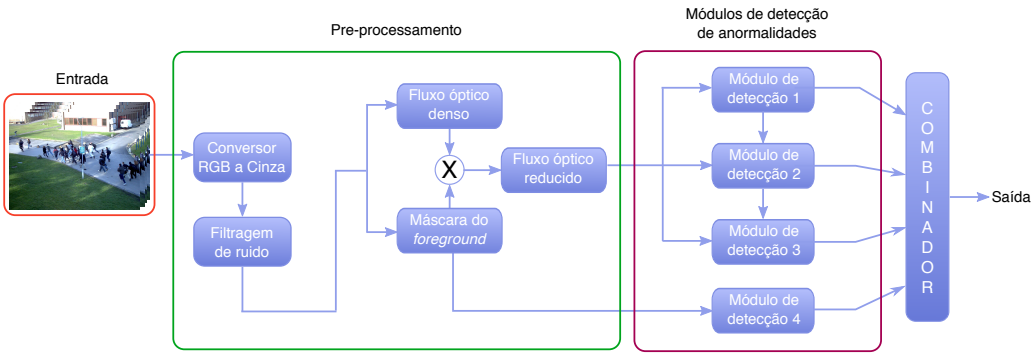


Figura 9 – *Arquitetura do sistema com módulo de pré-processamento detalhado.*

O fluxo óptico reduzido apresentado na Figura 8 é uma imagem, do mesmo tamanho da imagem de entrada, com os valores dos vetores de movimento codificados usando uma equivalência em cores. Entretanto, os algoritmos de fluxo óptico calculam os valores de deslocamento do pixel na direção horizontal e vertical, o que permite obter a magnitude e direção do deslocamento. Por conseguinte, a entrada dos módulos 1, 2 e 3 são duas matrizes de valores reais com dimensões iguais à imagem de entrada. A primeira matriz, denotada como $\boldsymbol{M}$ armazena os valores de magnitude. A segunda matriz, denotada como $\boldsymbol{D}$ armazena os valores de direção. De maneira que a magnitude e a direção de deslocamento do pixel (x, y) na imagem de entrada estão armazenados nos elementos $M(x, y)$ e $D(x, y)$ respectivamente.

A seguir serão detalhados os módulos de detecção de anormalidade.

3.2 Caso 1: Aumento repentino da velocidade

O primeiro tipo de anormalidade a ser considerada apresenta-se quando um grupo de pessoas ingressa ao campo visual da câmera caminhando e, repentinamente, todas começam a correr, aumentando significativamente sua velocidade média. Esse tipo de comportamento apresenta-se em situações onde acontece algo que assusta as pessoas em uma multidão, como uma explosão ou um acidente, e todas elas começam a correr.

A Figura 10 apresenta exemplos de situações que serão classificadas como anormais neste módulo.

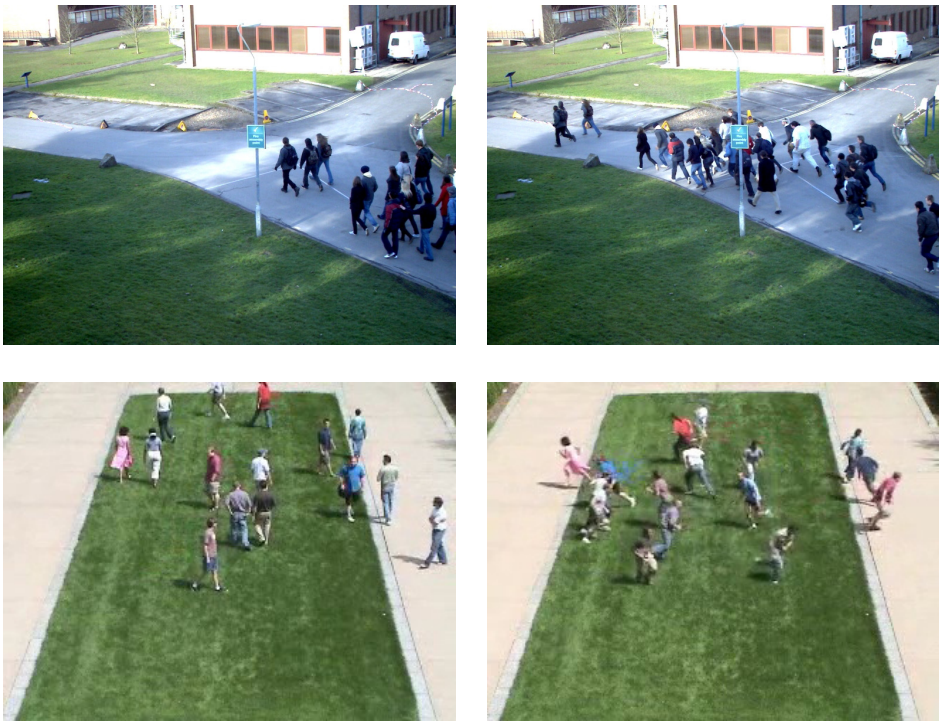


Figura 10 – *Exemplos de anormalidade detectadas no módulo 1. Na primeira coluna apresenta-se comportamento normal. Na segunda coluna apresenta-se comportamento anormal.*

Neste tipo de anormalidade existe uma mudança significativa nos valores atuais de velocidade com respeito a valores recentes. O objetivo é, portanto, identificar o momento no qual exista uma mudança muito grande nos valores médios na magnitude do fluxo óptico com respeito a valores anteriores. Uma técnica que visa realizar uma comparação semelhante foi proposta por Evan Ribnick e colaboradores em 2006 no artigo (RIBNICK *et al.*, 2006).

No artigo do Ribnick é apresentada uma técnica para detectar o momento no qual uma câmera de vídeo é ocluída por algum objeto. A técnica é baseada no princípio de que *quando uma câmera é ocluída os frames mais recentes do vídeo são significativamente diferentes daqueles frames mais antigos*. Esse mesmo princípio pode ser aplicado neste

tipo de anomalia, onde, devido à mudança na velocidade, a magnitude do movimento das pessoas nos *frames* recentes é muito diferente da magnitude nos *frames* antigos.

No trabalho do Ribnick são utilizados dois *buffers* que armazenam *frames* consecutivos do vídeo. Os dois *buffers* são estruturas de dados tipo FIFO (*First Input First Output*) sendo um deles com muita maior capacidade que o outro. O buffer maior contém o histórico antigo e o menor contém os eventos mais recentes. Ribnick utiliza três métricas: cor, intensidade e gradiente. As medianas de cada métrica são calculadas para os *frames* em cada *buffer* e posteriormente comparadas, assim se existir uma diferença entre eles maior do que um *threshold* fixo (estabelecido empiricamente) então o sistema detecta a oclusão da câmera.

O algoritmo proposto neste trabalho será baseado na ideia de Ribnick de utilizar dois *buffers* para comparar o histórico de magnitude do fluxo óptico de amostras atuais e recentes. Assim, na sequência será detalhado o algoritmo proposto.

Inicialmente são definidos os dois *buffers* do tipo FIFO com tamanhos fixos L e C que podem ser ajustados pelo usuário, levando em conta que $C \ll L$. O primeiro *buffer* de tamanho C é chamado de **fila curta** e é responsável pelo armazenamento de informação recente, enquanto o *buffer* de tamanho L é chamado de **fila longa** e armazena as informações passadas.

Seja $\boldsymbol{M}$ a matriz com os valores de magnitude do fluxo óptico reduzido obtido como foi detalhado na Seção 3.1. Primeiro é definido um vetor $\mathbf{m} = \{m_1, m_2, m_3, \ldots, m_n\}$ contendo os valores de $\boldsymbol{M}$ diferentes de zero. A seguir é calculada a média $\bar{m}$ de $\mathbf{m}$.

Uma diferença da técnica apresentada em (RIBNICK *et al.*, 2006) e a proposta neste trabalho é que o conteúdo dos *buffers* não são as imagens completas obtidas pela câmera senão a média das magnitudes diferentes de zero da matriz do fluxo óptico de cada *frame*, o que reduz a quantidade de memória necessária para o funcionamento do algoritmo.

Para cada novo *frame* o valor $\bar{m}$ é calculado e adicionado na fila curta de acordo com a seguinte sequência:

1. O último elemento da fila longa é eliminado.
2. O último elemento da fila curta é copiado à fila longa.
3. O último elemento da fila curta é eliminado.
4. O novo valor $\bar{m}$ é introduzido à fila curta.

Seguindo essa sequência o número de elementos nas duas filas sempre será o mesmo. Apresenta-se na Figura 11 uma descrição gráfica da sequência descrita anteriormente.

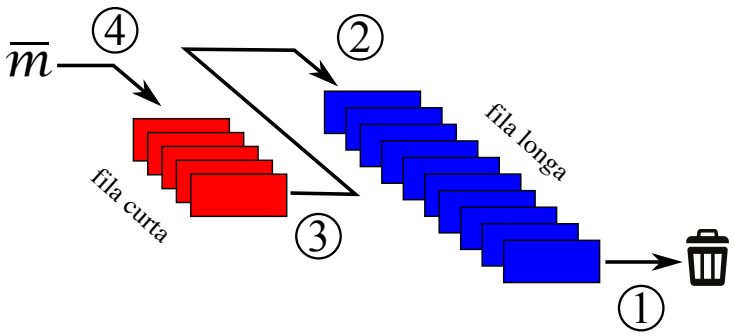


Figura 11 – *Sequência de preenchimento das duas filas.*

Inicialmente as duas filas estão vazias, quando é detectado um grupo de pessoas entrando na cena, é obtido o valor médio $\bar{m}$ e é armazenado na fila curta, entretanto a fila longa ainda não possui nenhuma informação, portanto são requeridos no mínimo $C + L$ *frames* para preencher as duas filas e o módulo começar a detectar possíveis anomalias.

Após completar os $C + L$ *frames* iniciais começa o processo de comparação das duas filas. Seja $\mathbf{f}_c = \{\bar{m}_1, \bar{m}_2, \bar{m}_3, \ldots, \bar{m}_C\}$ o vetor de valores contidos na fila curta e $\mathbf{f}_l = \{\bar{m}_{C+1}, \bar{m}_{C+2}, \bar{m}_{C+3}, \ldots, \bar{m}_{C+L}\}$ o vetor de valores contidos na fila longa. A cada novo valor $\bar{m}$ obtido é calculada a média (μ) e o desvio padrão (σ) dos valores contidos na fila longa $\mathbf{f}_l$ segundo,

$$\mu = \frac{1}{L} \sum_{i=C+1}^{C+L} \bar{m}_i \tag{3.1}$$

$$\sigma = \sqrt{\frac{1}{(L-1)} \sum_{i=C+1}^{C+L} (\bar{m}_i - \mu)^2} \tag{3.2}$$

A seguir é calculada a mediana ($\tilde{f}_c$) dos valores contidos na fila curta. É utilizada a mediana já que é uma medida estatística menos suscetível ao ruído do que a média. A mediana também permite reduzir os falsos positivos já que são necessários $C/2$ *frames* de comportamento anormal para a mediana mudar de valor.

A média e o desvio padrão da fila longa (μ, σ) são utilizados como parâmetros em uma função de distribuição de probabilidade normal $\mathcal{N}$ segundo,

$$\mathcal{N}(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-y)^2}{2\sigma^2}} \tag{3.3}$$

onde x é o valor da variável aleatória, neste caso $x = \bar{m}_i | \bar{m}_i \in \mathbf{f}_l$. Essa função de probabilidade Gaussiana é utilizada para determinar o momento no qual acontece o comportamento anormal. Dado que em uma distribuição de probabilidades Gaussiana o intervalo $[\mu - 3 * \sigma, \mu + 3 * \sigma]$ oferece um nível de confiança de pertença de 99% então é definido

o seguinte critério: se o valor $\tilde{f}_c$ (a mediana dos valores contidos em $\mathbf{f}_c$) é maior do que $\mu + 3 * \sigma$ e menor do que $\mu - 3 * \sigma$ então é dito que $\tilde{f}_c$ pertence ao modelo Gaussiano, caso contrário é dito que $\tilde{f}_c$ não pertence ao modelo e portanto é classificado como comportamento anormal.

A Figura 12 apresenta graficamente o critério de pertinência. Se o valor $\tilde{f}_c$ encontra-se dentro da região azul então é considerado comportamento normal, caso contrário, significa que ocorreu uma mudança significativa nos valores de magnitude nos últimos *frames* em comparação aos *frames* mais antigos e, portanto, é detectada uma anormalidade.

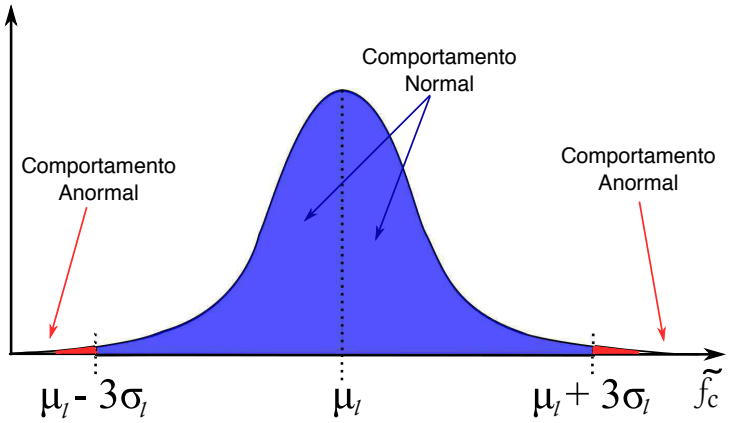


Figura 12 – *Região de pertinência (em azul) do valor $\tilde{f}_c$ na função de distribuição de probabilidade $\mathcal{N}(\mu, \sigma)$*

Se o valor $\tilde{f}_c$ estiver fora dos limites definidos pela Gaussiana a transferência de valores à fila longa é suspensa e o modelo Gaussiano não é atualizado até que os valores de $\tilde{f}_c$ voltem a pertencer ao último modelo Gaussiano obtido.

Com a abordagem aqui apresentada, o modelo é atualizado constantemente o que leva uma flexibilidade alta sem precisar utilizar algoritmos muito complexos. Os resultados dos testes realizados com este algoritmo serão apresentados na Seção 4.3.

3.3 Caso 2: Anormalidade na direção e/ou velocidade local

Neste módulo serão detectadas anormalidades associadas a variações significativas na velocidade ou direção de deslocamento em relação a um modelo obtido a partir de amostras unicamente com comportamento normal. Em contraste com o módulo anterior, neste módulo não é necessário que todas as pessoas na cena apresentem o mesmo comportamento anormal.

O uso de amostras de comportamento normal para a criação de um modelo que será posteriormente utilizado para detecção de *outliers*, aparece em vários trabalhos na literatura. No entanto, eles geram modelos fechados que não podem ser mais atualizados

uma vez terminada a fase de treinamento, impossibilitando a adição de novo conhecimento ao modelo. Na abordagem proposta neste trabalho é desejado que o modelo possa incluir novo conhecimento sem precisar treiná-lo a cada nova atualização.

Neste módulo as anomalias a serem detectadas estão divididas em três categorias: anormalidade por presença, anormalidade por direção e anormalidade por magnitude. A anormalidade por presença apresenta-se quando uma ou várias pessoas se deslocam em uma zona da imagem onde poucas ou nenhuma pessoa estive anteriormente. A anormalidade por direção apresenta-se quando uma ou várias pessoas se deslocam em uma direção pouco frequente em uma região específica da imagem. Finalmente a anormalidade por magnitude apresenta-se quando uma ou várias pessoas se deslocam com uma velocidade diferente (superior ou inferior) às observadas no modelo.

A Figura 13 apresenta exemplos de cada um desses comportamentos.

(a) (b) (c)

Figura 13 – *Exemplo de anormalidades detectadas neste módulo: (a) anormalidade por presença, (b) anormalidade por direção e (c) anormalidade por magnitude.*

A metodologia usada na detecção de cada uma das três categorias de anormalidade é diferente. No entanto as três categorias foram reunidas neste módulo por três razões: a primeira razão é que todas as categorias de anormalidade podem ser obtidas por meio da informação do fluxo óptico reduzido, portanto, as três categorias possuem a mesma entrada. A segunda razão é que este módulo se especializa na detecção de anormalidade local, ou seja, onde uma ou poucas pessoas apresentam o comportamento anormal; e os três tipos de anormalidade detectadas neste módulo são locais. E a terceira razão tem a ver com a implementação dos algoritmos de detecção. Embora a metodologia de detecção seja diferente para cada categoria, elas compartilham a ideia de localidade, portanto, nas três categorias as imagens são divididas em regiões retangulares iguais. Consequentemente, cada algoritmo processa as mesmas informações de cada região de forma diferente. Por causa da divisão nas três categorias, este módulo apresentará como saída três imagens binárias. A primeira contendo anormalidades por presença, a segunda anormalidade por direção e a terceira anormalidade por magnitude.

Na sequência será descrita a divisão por regiões a qual é comum nas três categorias e posteriormente será detalhado cada algoritmo de detecção.

As entradas deste módulo serão as matrizes $\boldsymbol{M}$ e $\boldsymbol{D}$ contendo os valores de magnitude e direção do fluxo óptico reduzido obtido segundo descrito na Seção 3.1.

O fluxo óptico é um algoritmo que apresenta alta sensibilidade ao ruído nas imagens de entrada, portanto é necessário realizar uma filtragem, tanto espacial quanto temporal. Para realizar dita filtragem são armazenadas as N últimas matrizes de magnitude $\boldsymbol{M}$ e de direção $\boldsymbol{D}$. A filtragem espacial é realizada aplicando um filtro gaussiano a cada uma das N matrizes. A filtragem temporal produz duas matrizes $\widetilde{\boldsymbol{M}}$ e $\widetilde{\boldsymbol{D}}$. O valor $\widetilde{M}_k(x, y)$ da matriz de magnitudes filtrada $\widetilde{\boldsymbol{M}}$ no *frame* k na posição (x, y) é dada pela mediana dos valores $[M_{k-1}(x, y), M_{k-2}(x, y), \ldots, M_{k-N}(x, y)]$ das N matrizes de magnitudes anteriores. O valor $\widetilde{D}_k(x, y)$ é obtido de forma semelhante: como a mediana das N matrizes de direção anteriores na posição (x, y).

Devido ao efeito de perspectiva, a magnitude de deslocamento das pessoas caminhando mais próximas a câmera é maior do que aquelas que estão mais afastadas, por isso é comum dividir a imagem em regiões retangulares de pequeno tamanho e assim obter o modelo de cada região separadamente. Por conseguinte, neste módulo as matrizes $\widetilde{\boldsymbol{M}}$ e $\widetilde{\boldsymbol{D}}$ serão divididas em N_r regiões R_i organizadas em uma grade de F_r linhas e C_r colunas. Adicionalmente cada região R_i é subdividida em um número N_{sr} de sub-regiões denotadas como Sr_i^j, organizadas em uma grade de F_{sr} linhas e C_{sr} colunas. A Figura 14 apresenta um exemplo de divisão da matriz $\widetilde{\boldsymbol{M}}$ com $F_r = 4$ e $N_{sr} = 16$ sendo $F_{sr} = 4$ e $C_{sr} = 4$. Na notação Sr_i^j o índice i representa o número da região e o índice j representa o número da sub-região dentro da região i.

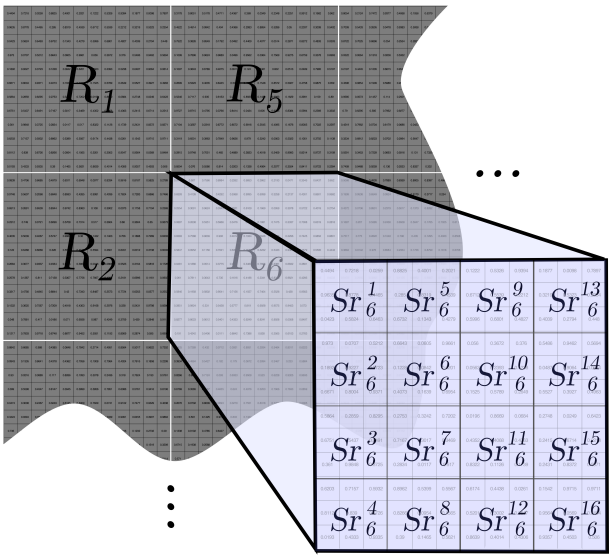


Figura 14 – *Divisão das matrizes de magnitude e direção em regiões R_i e em sub-regiões Sr_i^j.*

3.3.1 Anormalidade por presença

Nesta parte do módulo o objetivo é detectar pessoas se deslocando em uma zona da imagem onde existe baixa ou até nenhuma probabilidade de movimento das mesmas.

Para codificar e armazenar as informações de presença é proposto o uso de um histograma bidimensional. O histograma permite obter um modelo atualizável no tempo, possibilitando adicionar novas informações, manter o histórico passado e/ou eliminar informações pouco frequentes.

O histograma de presença bidimensional Hp é uma matriz de tamanho $F_p \times C_p$ onde F_p e C_p são o número de linhas e de colunas de Hp obtidas segundo as Equações 3.4 e 3.5 respectivamente.

$$F_p = F_r \times F_{sr} \tag{3.4}$$

$$C_p = C_r \times C_{sr} \tag{3.5}$$

onde F_r e C_r são o número de linhas e de colunas na grade de divisão das regiões; F_{sr} e C_{sr} são o número de linhas e de colunas na grade de subdivisão em cada região R_i. Assim, por exemplo, se a matriz $\widetilde{\boldsymbol{M}}$ é dividida em $N_r = 35$ regiões organizadas em $F_r = 5$ linhas e $C_r = 7$ colunas, e se cada região é dividida em $N_{sr} = 16$ sub-regiões organizadas em $F_{sr} = 4$ linhas e $C_{sr} = 4$ colunas, então o número de linhas F_p e de colunas C_p do histograma de presença é $F_p = 5 \times 4 = 20$ e $C_p = 7 \times 4 = 32$.

Inicialmente todos os elementos da matriz $\boldsymbol{H}_p$ são estabelecidos em zero. O valor $H_p(x, y)$ aumentará em uma unidade se é detectado algum movimento na sub-região Sr_i^j. Para obter as coordenadas (x, y) em $\boldsymbol{H}_p$ a partir dos índices i e j são utilizadas as Equações 3.6, 3.7 e 3.8.

$$ind_{ij} = (i - 1) * (F_{sr} * C_{sr}) + j \tag{3.6}$$

$$x = \left\lceil \frac{ind_{ij}}{F_p} \right\rceil \tag{3.7}$$

$$y = ind_{ij} - (x - 1) * F_p \tag{3.8}$$

A fim de evitar que as barras do histograma $\boldsymbol{H}_p$ aumentem indefinidamente, é estabelecido um valor de altura máxima P_{max}, valor a partir do qual não é mais acrescentada a altura da barra no histograma. Além de P_{max} também é definido um parâmetro

de sensibilidade $\xi_p \in [0, 1]$ que permite calcular a altura mínima necessária das barras do histograma para considerar anormalidade por presença como

$$P_{min} = \xi_p * P_{max} \tag{3.9}$$

Dessa forma, se existir algum valor diferente de zero na sub-região Sr_i^j e o valor da altura da barra correspondente $H_p(x, y)$ for menor que P_{min}, então é detectada uma anormalidade por presença. Finalmente é criada uma imagem binária, do mesmo tamanho que $\widetilde{\boldsymbol{M}}$, denotada por $\boldsymbol{\mathcal{P}}$, contendo uns nas sub-regiões onde foram encontradas anormalidades.

A atualização do modelo é realizada através do histograma $\boldsymbol{H}_p$. A cada determinado número de *frames* é decrementada em uma unidade a altura de todas as barras do histograma. Como resultado, só serão classificadas como normais zonas da imagem que apresentem movimentação de forma consistente no tempo.

3.3.2 Anormalidade por direção

Para detectar anormalidades na direção de deslocamento das pessoas, é criado um modelo para cada região R_i usando os valores da matriz de direções filtrada $\widetilde{\boldsymbol{D}}$.

A Figura 15 apresenta um exemplo dos valores de direção da matriz obtidos para uma região R_i em um único *frame*. Da Figura 15 é possível verificar que, embora a pessoa esteja caminhando em uma única direção, existem vetores com valores muito dispersos devido ao movimento de suas pernas e pés. No entanto existe uma maior frequência entre os valores de 140 e 150 graus. Portanto, é desejado que o modelo mantenha esses valores mais frequentes enquanto ignora valores com frequências muito baixas.

Nesta parte do módulo será utilizado um modelamento baseado em probabilidades. Serão levadas em conta duas probabilidades: a probabilidade de uma determinada direção acontecer e a probabilidade condicional de uma determinada média de magnitude acontecer dada uma direção.

O algoritmo será detalhado para uma região em particular, mas o processo é similar nas regiões restantes. No primeiro passo são obtidos e armazenados no vetor $\tilde{\boldsymbol{d}}$ os valores de direção diferentes de zero da matriz $\widetilde{\boldsymbol{D}}$ contidos em uma região R em particular segundo a seguinte equação

$$\tilde{\boldsymbol{d}} = \left\{ \widetilde{D}(x, y) | (x, y) \in R, \widetilde{D}(x, y) \neq 0 \right\} \tag{3.10}$$

A seguir é calculado o histograma $\boldsymbol{H}$ de valores em $\tilde{\boldsymbol{d}}$ utilizando um intervalo de

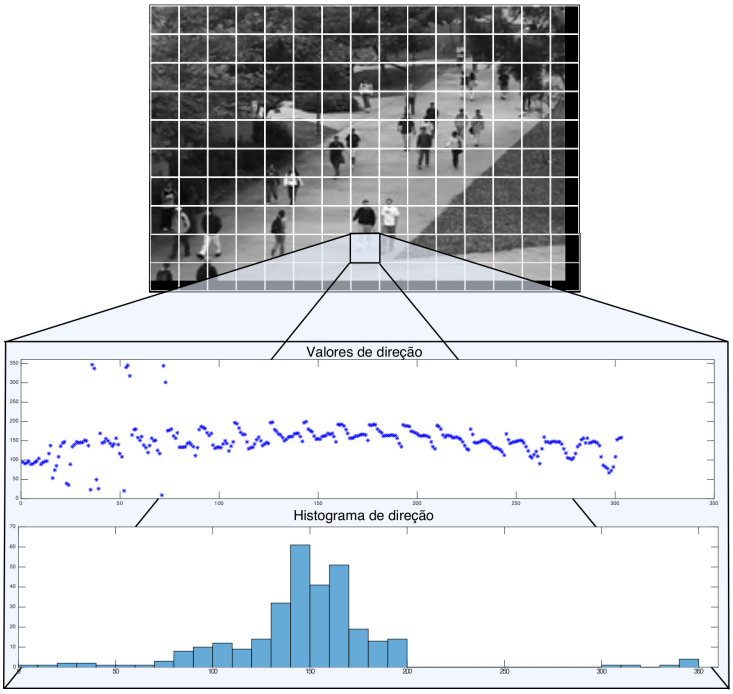


Figura 15 – *Exemplo dos valores de direção dos vetores de fluxo óptico e seu histograma para uma região R_i.*

quantização $\Delta\theta$ obtendo assim η barras, com η calculado segundo a Equação 3.11.

$$\eta = \left\lceil \frac{360}{\Delta\theta} \right\rceil \tag{3.11}$$

Em seguida é realizada uma filtragem das barras no histograma com baixa frequência segundo a seguinte condição,

$$H(x) = \begin{cases} 0, & \text{se } H(k) < \gamma \\ H(k), & \text{caso contrario} \end{cases} \tag{3.12}$$

onde γ é um parâmetro de filtragem. A seguir será obtida a soma das magnitudes dos vetores de fluxo óptico cujas direções estejam dentro de cada intervalo k segundo a Equação 3.13

$$m(k) = \sum \widetilde{M}\,(x,y) | (x,y) \in R, (k-1)\Delta\theta < \widetilde{D}(x,y) \leq k\Delta\theta, H(k) \neq 0 \;\forall\; k \in [1,\eta]. \tag{3.13}$$

Finalmente, a dupla de valores k e $\overline{m}(k)$ é armazenada, onde $\overline{m}(k)$ é obtida como.

$$\overline{m}(k) = \frac{m(k)}{H(k)} \;\forall\; H(k) \neq 0 \tag{3.14}$$

Após armazenar todas as duplas k e $\overline{m}(k)$ obtidas usando os vídeos de treinamento é calculado um histograma $\boldsymbol{H}_D$ de todos os k valores armazenados. O histograma $\boldsymbol{H}_D$ contém a frequência com que cada direção k aparece nos vídeos de treinamento. A seguir são gerados os modelos de probabilidade. Seja $\boldsymbol{K}$ a variável aleatória discreta de posição no histograma $\boldsymbol{H}_D$, e $\overline{\boldsymbol{M}}$ a variável aleatória contínua da média de magnitude dos vetores de fluxo óptico. Portanto é definida a probabilidade conjunta de uma região apresentar fluxo óptico com direção dentro do intervalo k e contendo uma média de magnitude $\overline{m}$ como

$$P(\boldsymbol{K} = k, \overline{\boldsymbol{M}} = \overline{m}) = P(\overline{\boldsymbol{M}} = \overline{m}, \boldsymbol{K} = k) * P(\boldsymbol{K} = k) \tag{3.15}$$

A probabilidade marginal de ocorrência da direção $P(\boldsymbol{K} = k)$ é obtida como

$$P(\boldsymbol{K} = k) = \frac{H_D(k)}{\sum \boldsymbol{H}_D} \tag{3.16}$$

Enquanto que é usada uma distribuição de probabilidade Gaussiana para obter a probabilidade condicional $P(\overline{\boldsymbol{M}} = \overline{m}, \boldsymbol{K} = k)$ como

$$P(\overline{\boldsymbol{M}} = \overline{m}, \boldsymbol{K} = k) = \mathcal{N}(\overline{m}, \mu_k, \sigma_k) \tag{3.17}$$

onde $\mathcal{N}(\overline{m}, \mu_k, \sigma_k)$ é a função de distribuição de probabilidade Gaussiana definida na Equação 3.3. Os parâmetros μ_k e σ_k são a média e o desvio padrão de todos os valores $\overline{m}$ contidos no intervalo $\boldsymbol{K} = k$.

Assim, para determinar se existe anormalidade por direção em uma região é comparada a probabilidade do fluxo óptico apresentar uma determinada direção k e uma determinada magnitude $\overline{m}$ usando um valor de *threshold* T_D fixo . Caso $P(\boldsymbol{K} = k, \overline{\boldsymbol{M}} = \overline{m})$ for menor do que T_D é dito que os vetores com direção dentro do intervalo k apresentam anormalidade, caso contrário são classificados como normais.

Visando manter o modelo atualizado e eliminar direções muito pouco frequentes, após um número fixo de *frames* podem ser eliminadas da lista uma ou varias duplas k e $\overline{m}(k)$.

Este algoritmo retorna uma imagem binária do mesmo tamanho que $\widetilde{\boldsymbol{M}}$, denotada como $\boldsymbol{\mathcal{F}}$, contendo uns nas posições onde foram detectadas anormalidades por direção.

3.3.3 Anormalidade por velocidade

No desenvolvimento do algoritmo para identificação de anormalidades por velocidade foram encontrados alguns aspectos de desenho que deviam ser levados em conside-

ração para obter o desempenho desejado. Por exemplo, o modelo de magnitude devia ser local, já que diferentes zonas da imagem apresentam diferentes padrões de movimento. Similarmente, desejava-se que o modelo tivesse a capacidade de codificar diferentes padrões de movimento em uma mesma região da imagem, por exemplo quando uma estrada é compartilhada por carros, pessoas e bicicletas, cada uma delas com um padrão de velocidade diferente. Também foi encontrada a necessidade de distinguir entre movimento horizontal e vertical, devido ao fato do movimento horizontal na imagem apresentar velocidade aparente maior do que o movimento vertical. Por último desejava-se que o modelo possuísse um mecanismo de atualização, onde novos padrões de movimento pudessem ser incluídos e padrões muito pouco frequentes pudessem ser excluídos.

Visando cumprir esses aspectos é proposta uma modelagem baseada em **templates**, **mapas de magnitudes**, uma medida de similaridade e uma medida de frequência.

O algoritmo aqui descrito foca-se no processamento da magnitude do fluxo óptico. No entanto, a matriz de direção $\widetilde{\boldsymbol{D}}$ é utilizada para determinar a direção principal de deslocamento.

Neste algoritmo é utilizada a divisão das matrizes $\widetilde{\boldsymbol{M}}$ e $\widetilde{\boldsymbol{D}}$ em regiões como foi descrito no final da Seção 3.3. A modelagem e detecção de anormalidade são realizadas em cada região independentemente, portanto será descrito o algoritmo para uma região em particular, mas o mesmo processo se aplica para as restantes.

O primeiro passo do algoritmo é identificar o sentido principal do movimento para classificá-lo como sendo horizontal ou vertical. Para lograr isso, os valores de direção diferentes de zero dos vetores de fluxo óptico contidos na região R_i são quantizados utilizando um intervalo de quantização $\Delta\theta$. A seguir é construído um histograma onde o centro de cada barra é o valor de direção quantizado e a altura é a soma das magnitudes de cada um dos vetores contidos na barra. Matematicamente, a altura $h(k)$ da barra k é calculada através da Equação 3.18.

$$h(k) = \sum_{(x,y)\in R_i} w_k(x,y) * \widetilde{M}(x,y) \tag{3.18}$$

onde o valor w_k é calculado a partir de

$$w_k(x,y) = \begin{cases} 1, & \text{se } (k-1)\Delta\theta < \widetilde{D}(x,y) \leq k\Delta\theta \\ 0, & \text{caso contrario} \end{cases} \tag{3.19}$$

para $k = \left\{1, \ldots, \frac{360}{\Delta\theta}\right\}$. A seguir é obtida a barra $\overline{k}$ com maior altura. Finalmente o sentido de movimento é determinado como horizontal se $\overline{k}$ é igual a 1, 6, 7 ou 12, ou como vertical nos restantes casos. A Figura 16 apresenta os intervalos considerados como horizontais e

verticais.

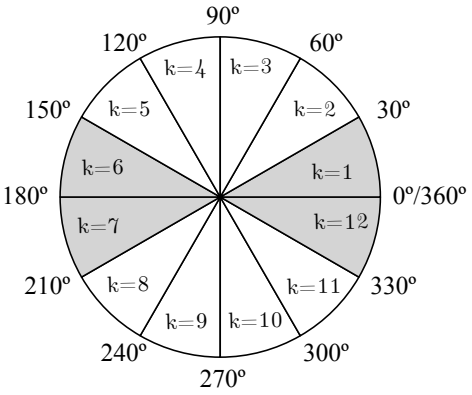


Figura 16 – *Quantização e determinação do sentido principal de movimento. Em cinza intervalos considerados com sentido horizontal e em cor branca intervalos considerados com sentido vertical.*

Para cada região serão obtidos dois modelos de magnitude: um para o sentido horizontal e outro para o sentido vertical.

A Figura 17 resume o processo de modelagem de magnitude de uma região em particular. A seguir serão detalhados seus componentes.

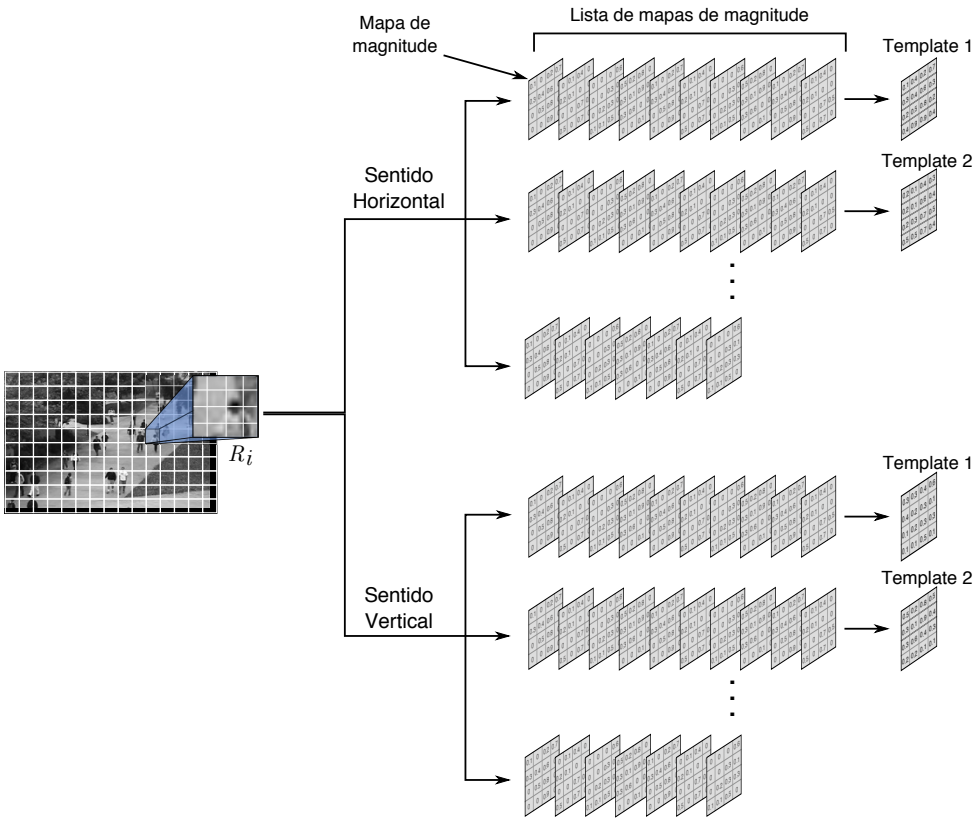


Figura 17 – *Modelagem da magnitude para uma região R_i de exemplo.*

O modelo de magnitude é composto de um conjunto de matrizes chamadas ***templates***. Os *templates* são gerados por meio de uma série de outras matrizes chamadas de **mapas de magnitude**.

Uma vez determinado o sentido do movimento são construídos os mapas de magnitude. Os mapas de magnitudes são matrizes de tamanho $F_{sr} \times C_{sr}$, onde F_{sr} e C_{sr} são o número de linhas e o número de colunas da grade gerada pelas sub-regiões de R_i respectivamente. Os mapas de magnitude são construídos a cada novo *frame* ao obter a mediana dos valores de magnitude diferentes de zero na matriz $\widetilde{\boldsymbol{M}}$ contidos em cada sub-região. Dessa forma, para obter os elementos de um mapa de magnitude primeiro é obtido o vetor $\boldsymbol{m}_i^j$ contendo os valores diferentes de zero em $\widetilde{\boldsymbol{M}}$ pertencentes à região i e à sub-região j, segundo

$$\boldsymbol{m}_i^j = \left\{ \widetilde{M}(x,y) |\, \widetilde{M}\,(x,y) \neq 0, (x,y) \in Sr_i^j \right\} \tag{3.20}$$

A seguir os elementos do mapa de magnitudes $\boldsymbol{\mathcal{S}}_i$ da região R_i são obtidos usando a Equação 3.21

$$\mathcal{S}_i(f,c) = \widetilde{m}_i^j \tag{3.21}$$

onde $\widetilde{m}_i^j$ é a mediana dos valores contidos em $\boldsymbol{m}_i^j$; f e c são o número da linha e da coluna respectivamente e são obtidos a partir do número da sub-região j usando as Equações 3.22 e 3.23.

$$c = \left\lceil \frac{j}{F_{sr}} \right\rceil \tag{3.22}$$

$$f = j - (c-1) * F_{sr} \tag{3.23}$$

Os mapas de magnitude são armazenados em uma lista com um tamanho fixo L_m. Após o número mínimo L_m de mapas de magnitude ter sido atingido, é gerado um *template*. Um *template* é uma matriz de tamanho $F_{sr} \times C_{sr}$ construída a partir dos mapas de magnitudes armazenados na lista correspondente. Assim, seja $\boldsymbol{\mathcal{T}}_i$ um *template* da região R_i. O elemento $\mathcal{T}_i(f,c)$ é obtido como a média aritmética dos valores na posição (f,c) de todos os mapas de magnitudes contidos na correspondente lista de mapas. Se, por acaso, os elementos na posição (f,c) forem zero em todos os mapas de magnitude, o valor $\mathcal{T}_i(f,c)$ é estabelecido como a média dos restantes elementos de $\boldsymbol{\mathcal{T}}_i$. Uma vez construído o *template* os novos mapas de magnitudes não serão armazenados na lista.

Os *templates* são utilizados para determinar se um novo mapa de magnitude é classificado como normal ou anormal. Para realizar a mencionada classificação são comparados os valores contidos na matriz do *template* e os valores do novo mapa de magnitude.

Dita comparação é realizada por meio de uma métrica de distância elemento a elemento, portanto as duas matrizes $\boldsymbol{\mathcal{T}}_i$ e $\boldsymbol{\mathcal{S}}_i$ são vetorizadas gerando $\mathbf{t}_i$ e $\mathbf{s}_i$ respectivamente. Existem várias métricas para comparar dois vetores. Um exemplo delas é a norma Euclidiana definida na Equação 3.24.

$$d(\mathbf{t}_i, \mathbf{s}_i) = \sqrt{\sum_{n=1}^{N_{sr}} (\mathcal{T}_i(n) - \mathcal{S}_i(n))^2} \tag{3.24}$$

A métrica de distância $d(\mathbf{t}_i, \mathbf{s}_i)$ é comparada com um valor de *threshold* T_i fixo. É dito que o mapa de magnitude $\boldsymbol{\mathcal{S}}_i$ apresenta anormalidade se, e somente se, $d(\mathbf{t}_i, \mathbf{s}_i) > T_i$.

Até o momento foi descrita a geração de um único *template*, mas, como já foi mencionado, o modelo de magnitude é composto por vários *templates*, a seguir será detalhado o processo de criação dos *templates* restantes assim como a obtenção do valor T_i.

Inicialmente é definida uma primeira lista $\boldsymbol{\lambda}_i^1$ de mapas de magnitude vazia com um tamanho fixo, onde serão armazenados os primeiros mapas de magnitude.

Após a primeira lista conter L_m mapas de magnitude é obtido o primeiro *template* $\boldsymbol{\mathcal{T}}_i^1$ como foi detalhado anteriormente. A seguir é obtido o valor de *threshold* T_i como a distância máxima (segundo alguma métrica de distância adotada) entre o *template* $\boldsymbol{\mathcal{T}}_i^1$ e todos os mapas contidos na lista $\boldsymbol{\lambda}_i^1$. Após o *frame* $L_m + 1$ são obtidos novos mapas de magnitude que não são mais incluídos na lista $\boldsymbol{\lambda}_i^1$. Em vez disso, o novo mapa $\boldsymbol{\mathcal{S}}_i$ é comparado diretamente com o *template* $\boldsymbol{\mathcal{T}}_i^1$, usando a métrica de distância. Se a distância entre $\boldsymbol{\mathcal{T}}_i^1$ e $\boldsymbol{\mathcal{S}}_i$ for menor que T_i então o mapa de magnitude é classificado como normal, se, pelo contrário, for maior que T_i então é classificado como anormal e uma segunda lista $\boldsymbol{\lambda}_i^2$ é criada e o mapa de magnitude é adicionado nela. Para incluir um novo mapa subsequente na segunda lista deve-se cumprir que pelo menos exista uma distância entre o novo mapa e os componentes da lista menor do que T_i. Caso todas as distâncias sejam maiores que T_i então é criada uma nova lista e o mapa é adicionada a ela. O processo é repetido para todos os *frames* do vídeo. Cada nova lista criada armazenará L_m mapas de magnitudes, quando alguma das novas listas estiver completa, um novo *template* é gerado.

O mecanismo de criação de listas e *templates* descrito anteriormente serve para adicionar novos padrões de magnitude ao modelo, acrescentando novas informações constantemente. Com o objetivo de eliminar padrões criados como consequência de ruído, ou aqueles que aparecem com muita pouca frequência, é utilizado um mecanismo de frequências similar ao utilizado nas secções anteriores. Assim cada *template* tem associado um valor de frequência de ocorrência que aumenta em uma unidade quando um mapa de magnitude apresenta uma distância inferior ao *threshold*. São definidos três valores: M_{min}, M_{max} e ξ_m. O parâmetro M_{max} estabelece o valor máximo que pode apresentar a frequência de ocorrência e $\xi_m \in [0, 1]$ é o parâmetro de sensibilidade. Os parâmetros M_{max} e ξ_m

são estabelecidos pelo usuário. O valor M_{min} é calculado segundo a Equação 3.25.

$$M_{min} = \xi_m * M_{max}; \tag{3.25}$$

Assim, existem dois casos onde um mapa de magnitudes é classificado como anormal: o primeiro acontece quando, ao calcular a distância entre o mapa atual e todos os *templates* do modelo, não existe nenhum valor inferior a T_i; e o segundo caso acontece quando, existindo um *template* com valor de distância inferior a T_i, a frequência de ocorrência associada aquele *template* apresenta um valor inferior a M_{min}.

Finalmente, quando um mapa de magnitude é classificado como anormal, são criados intervalos de pertinência de magnitude a partir de todos os *templates* existente para a correspondente direção, ditos intervalos determinarão quais vetores de fluxo óptico dentro da região apresentam magnitudes anormais e quais não. Portanto se em um determinado *frame* o mapa $\boldsymbol{\mathcal{S}}_i$ é classificado como anormal, e se N_T é o número de *templates* nessa região com frequência de ocorrência superior a M_{min}, então o conjunto de intervalos de pertinência é definido como

$$\{[\mu_i - 3 * \sigma_i, \mu_i + 3 * \sigma_i]\} \; \forall \, i \in [1, N_T] \tag{3.26}$$

onde μ_i é a média e σ_i é o desvio padrão dos valores do *template* $\boldsymbol{\mathcal{T}}_i$. Finalmente é definida uma matriz do mesmo tamanho que a região R_i contendo uns na posição (x, y) onde $\tilde{M}(x, y)$ não esteja contido em nenhum intervalo de pertinência definidos anteriormente.

Realizando o processo anterior para cada uma das regiões R_i é obtida uma imagem binária de anormalidade de magnitude, denotada como $\boldsymbol{\mathcal{B}}$ com 1's nos pixels anormais e zeros nos normais.

Os resultados obtidos ao implementar os algoritmos descritos nesta seção serão apresentados na Seção 4.4.

3.4 Caso 3: Movimentação na direção contrária à da multidão

Neste terceiro módulo serão identificadas anormalidades que acontecem quando uma, ou poucas pessoas, deslocam-se na direção contrária à direção de deslocamento de uma multidão. A Figura 18 apresenta exemplos desses cenários. É importante identificar esse tipo de anormalidade porque ela pode gerar situações de risco tanto para a pessoa indo em direção contrária quanto para a multidão.

Embora o módulo foca-se no processamento da direção, as informações de magnitude são utilizadas para gerar uma máscara inicial, portanto a entrada deste módulo são as matrizes $\boldsymbol{M}$ e $\boldsymbol{D}$ definidas na Seção 3.1

(a) (b)

Figura 18 – *Exemplos de anormalidade detectada neste módulo: (a) uma pessoa indo em direção contrária à multidão e (b) uma pessoa tentando ingressar em um prédio quando todas as outras estão saindo.*

O objetivo deste algoritmo é agrupar valores de direção semelhantes. Esse agrupamento permitiria detectar quando vetores muito diferentes aparecem no centro da multidão. Em contraste com os dois primeiros módulos, neste módulo não existirá um modelo construído a partir de amostras consideradas como normais. Ao invés disso, cada *frame* será analisado sem levar em conta um histórico recente dos mesmos.

Neste módulo unicamente serão levados em conta cenários que apresentem multidões, portanto é necessário ignorar aqueles *frames* com pouca densidade de pessoas. Para determinar se um *frame* em particular apresenta uma multidão, é obtida uma imagem binária $\boldsymbol{G}$ a partir da matriz $\boldsymbol{M}$ segundo a Equação 3.27.

$$G(x,y) = \begin{cases} 1, & \text{se } M(x,y) > 0 \\ 0, & \text{caso contrario} \end{cases} \tag{3.27}$$

A seguir são obtidos os componentes conexos da imagem $\boldsymbol{G}$, ignorando componentes com área inferior a uma certa porcentagem, denotada por ϕ, da área total do *frame*. Assim, por exemplo, se o *frame* de entrada tem um tamanho de 360x240 pixels, sua área é de 86.400 pixels, e se é estabelecida uma porcentagem de filtragem de 40%, significa que componentes conexos com áreas inferiores a 34.560 pixels serão ignorados pelo algoritmo por não serem considerados como multidão.

Se um determinado componente conexo cumpre com o mínimo de área para ser processado, o passo a seguir é obter sua direção mais frequente. Para isso são obtidos os valores de direção da matriz $\boldsymbol{D}$ nas posições definidas pelo componente conexo. Matematicamente, se $\boldsymbol{g}_k$ é o k-ésimo componente conexo de $\boldsymbol{G}$ que cumpre a condição de área mínima, então o vetor $\boldsymbol{d}$ conterá os elementos em $\boldsymbol{D}$ que pertencem a $\boldsymbol{g}_k$ segundo a

Equação 3.28

$$\boldsymbol{d} = \{D(x,y)|(x,y) \in \boldsymbol{g}_k\} \tag{3.28}$$

A seguir os valores contidos em $\boldsymbol{d}$ são quantizados usando um intervalo de quantização $\Delta\theta$ para seguidamente construir um histograma com η barras. O número de barras η é obtido usando a Equação 3.11. No passo seguinte é obtida a barra com maior altura, ou seja, o intervalo de direção mais frequente do componente conexo. Seja $\hat{\theta}$ a barra com maior frequência. A partir de $\hat{\theta}$ é definido um intervalo de ângulos anormais $\boldsymbol{\theta}_{ano}$ como sendo o conjunto de ângulos entre $\hat{\theta} + 135^o$ e $\hat{\theta} + 225^o$ como é ilustrado na Figura 19.

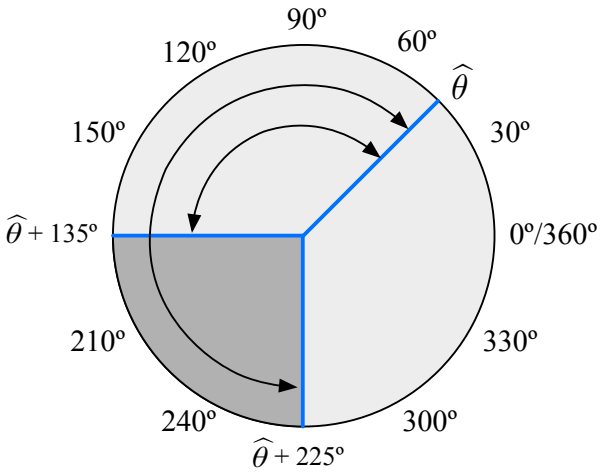


Figura 19 – *Exemplo de intervalo de ângulos anormais. Se $\hat{\theta}$ é o ângulo mais frequente em um determinado componente conexo, os ângulos anormais serão aqueles que estejam contidos no intervalo $[\hat{\theta} + 135^o, \hat{\theta} + 225^o]$ (região em cinza escuro na figura).*

A seguir é construída uma imagem binária $\boldsymbol{\mathcal{W}}$ contendo uns nas posições onde existe um valor de direção que pertence ao intervalo de anormalidade segundo a Equação 3.29.

$$\mathcal{W}(x,y) = \begin{cases} 1, & \text{se } \hat{\theta} + 135^o \leq D(x,y) \leq \hat{\theta} + 225^o \\ 0, & \text{caso contrario} \end{cases} \tag{3.29}$$

Os resultados obtidos ao implementar o algoritmo descrito nesta seção são apresentados na Seção 4.5.

3.5 Caso 4: Anormalidade por aparência

Neste último módulo serão obtidos modelos de aparência diretamente da imagem de *foreground*, tornando-o o único módulo que não leva em conta as informações do fluxo

óptico.

Exemplos de situações que visam ser identificadas como anormalidades neste módulo são apresentadas na Figura 20.

(a) (b)

Figura 20 – *Exemplos de anormalidade por aparência.*

De acordo com a revisão da literatura apresentada no Capítulo 2, só recentemente foram desenvolvidos algoritmos para detectar anormalidades tanto de magnitude quanto de aparência. Nos trabalhos que incluem detecção de anormalidade por aparência são utilizadas duas técnicas: texturas dinâmicas (TD) (MAHADEVAN *et al.*, 2010) e histogramas de orientações do gradiente (HOG) (DALAL; TRIGGS, 2005). Os trabalhos que utilizam a técnica de HOG apresentam melhores resultados que aqueles que utilizam DT, no entanto, a técnica de HOG utiliza uma SVM de uma classe para classificar vetores *features* que podem conter centenas ou até milhares de elementos. Nos trabalhos que utilizam HOG a SVM é treinada com amostrar consideradas normais e, após o treinamento, o modelo não é mais atualizado impedindo adicionar novas informações.

Visando reduzir a dimensão do vetor de *features* e obter um modelo atualizável é proposto o algoritmo descrito na sequência.

O algoritmo proposto é parcialmente baseado no HOG presente na literatura, no entanto reduzindo significativamente o tamanho do vetor de *features*. A entrada do algoritmo é a máscara de *foreground* obtida no pré-processamento. Inicialmente é obtida uma grade de regiões retangulares de tamanho fixo, similares às obtidas no Módulo 2, denotadas como R_i. A seguir são obtidos os componentes conexos a partir da máscara de *foreground*. Cada componente conexo será denotado como B_k. Na sequência é realizada uma filtragem de B_k de acordo com a seguinte regra: se r_i denota a área da região R_i e b_k^i é o número de elementos de B_k contidos na região R_i então o componente conexo B_k é analisado se, e somente se, b_k^i cumpre: $\alpha_1 r_i \leq b_k^i \leq \alpha_2 r_i$ onde $\alpha_1, \alpha_2 \in [0, 1]$ são a porcentagem mínima e máxima da área da região R_i respectivamente sendo $\alpha_1 < \alpha_2$. Se o componente conexo B_k cumpre a condição anterior então é calculado seu gradiente por meio do operador Sobel. O operador Sobel obtém uma aproximação da magnitude e da direção do gradiente em cada pixel da imagem. A seguir é obtido o histograma de direções do gradiente para o componente B_k. É possível definir a resolução de dito histograma a

partir do número de barras que o histograma conterá, denotado por η. Alternativamente, pode-se obter o número de barras a partir do intervalo de quantização $\Delta\theta$ (Equação 3.11).

Quanto menor for $\Delta\theta$ maior será o tamanho do histograma e mais detalhes ele armazenará. Para classificar novos histogramas como sendo normais ou anormais, é utilizada uma métrica de distância entre histogramas junto com uma medida de probabilidade de ocorrência.

A Figura 21 apresenta o processo de obtenção do histograma de direções do gradiente.

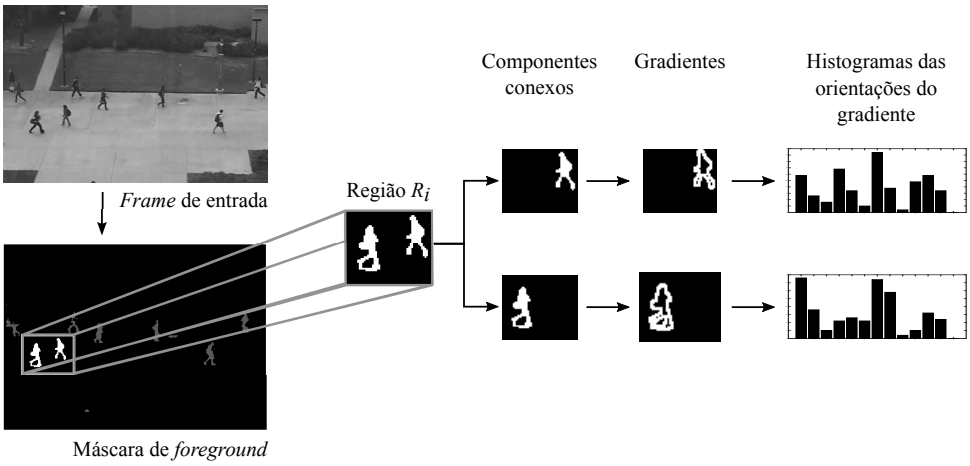


Figura 21 – *Exemplo da obtenção dos histogramas de orientações dos gradientes utilizados como features neste módulo.*

Para cada região R_i é definido um modelo de aparência e cada modelo está composto pelos seguintes elementos: uma matriz $\boldsymbol{P}_i$, a qual armazenará a lista de histogramas de direção; um vetor de frequências $\boldsymbol{D}_i$ e um valor de *threshold* de distância máxima T_a, que será o mesmo para todas as regiões. Inicialmente é definido o valor T_a, dito valor dependerá da métrica de distância utilizada. Em seguida, se um componente conexo cumpre a condição de pertinência para a região atual, então seu histograma de direções é obtido e armazenado na matriz $\boldsymbol{P}_i$ (inicialmente vazia). A partir do primeiro histograma armazenado, os histogramas consecutivos serão armazenados em $\boldsymbol{P}_i$ se, e somente se, não existir nenhuma distância entre o novo histograma e todos os histogramas contidos em $\boldsymbol{P}_i$ inferior a T_a. Caso exista alguma distância inferior a T_a, é obtido o modelo com menor distância e é acrescentada uma unidade ao valor de frequência na posição correspondente em $\boldsymbol{D}_i$.

Se a métrica de distância escolhida for a norma euclidiana então este algoritmo pode-se visualizar como um algoritmo de *clustering*, onde os histogramas em $\boldsymbol{P}_i$ representam centros de hiperesferas em uma dimensão η, T_a representa o raio dessa esfera e $\boldsymbol{D}_i$ representa a frequência com que um novo histograma é associado a cada esfera definida em $\boldsymbol{P}_i$.

No seguinte passo são obtidas as distâncias entre o novo histograma e todos aqueles em $\boldsymbol{P}_i$. Em seguida é obtida a linha j onde se encontra o histograma em $\boldsymbol{P}_i$ que gerou a menor dessas distâncias, denotada como $\hat{d}$. Posteriormente é obtida a probabilidade de ocorrência do histograma na linha j como

$$\hat{p}_i = \frac{D_i(j)}{\sum \boldsymbol{D}_i} \tag{3.30}$$

No último passo é obtido a fração $\phi = \hat{p}_i/\hat{d}$. O valor ϕ será baixo se a probabilidade de ocorrência do histograma é baixa ou se a distância $\hat{d}$ é alta. A classificação do componente conexo como normal ou anormal é, portanto, determinada pelo valor ϕ. Se ϕ for menor que um valor de *threshold* T_ϕ então o componente conexo é considerado anormal, caso contrário, é considerado normal.

Em síntese os parâmetros a serem estabelecidos neste algoritmo são: o tamanho da região R_i, a máxima distância T_a e o valor T_ϕ.

A inclusão de novas informações assim como a eliminação de histogramas pouco frequentes do modelo é possível graças ao vetor de frequências $\boldsymbol{D}_i$ que é atualizado a cada novo *frame*.

Os resultados obtidos ao implementar os algoritmos descritos nesta seção serão apresentados na Seção 4.6.

CAPÍTULO 4

Implementação e resultados

Neste capítulo serão apresentados os resultados obtidos ao implementar os algoritmos de detecção de anormalidade propostos no capítulo anterior. Os algoritmos foram implementados utilizando a ferramenta MatLab® na sua versão 2016a. Nos experimentos foram utilizados *datasets* contendo várias situações de anormalidade em diferentes cenários detalhados na sequência. As métricas de desempenho empregadas serão detalhadas na Seção 4.2.

4.1 *Datasets* utilizados

Não existe na literatura um único *dataset* que contenha todas as situações que pretendem ser detectadas neste trabalho, portanto, foram utilizados vários *datasets* alguns apresentando um único tipo de comportamento anormal enquanto outros contêm dois ou mais. Os diferentes *dataset* fornecem vídeos com diferentes resoluções, alguns desses vídeos estão em espaço de cor RGB e outros em tons de cinza. Como parte do pré-processamento das imagens de entradas os *frames* de todos os vídeos são redimensionados para garantir uma largura de 300 pixels, a altura de cada *frame* é obtida a partir da relação de aspecto da imagem original, mantendo as proporções originais. Além do anterior, todas as imagens coloridas são convertidas a tons de cinza.

4.1.1 PETS 2009 *dataset*

Os vídeos contidos no *dataset* PETS 2009 (PETS, 2009) estão divididos em quatro categorias: vídeos de treinamento (*S0-Training Data*), vídeos para estimação de densidade (*S1-Density Estimation*), vídeos para seguimento de pessoas (*S2-People Tracking*) e vídeos para o reconhecimento de eventos (*S3-Event Recognition*). Nos experimentos realizados neste módulo foram utilizados unicamente os vídeos da categoria de reconhecimento de

eventos. A Figura 22 apresenta *frames* de exemplo com comportamento normal e anormal. Os vídeos deste *dataset* possuem uma resolução de 768x576 pixels e as imagens são coloridas.

Os vídeos deste *dataset* proporcionam anormalidades nas Situações 1 e 3.

Figura 22 – *Frames de exemplo com comportamento normal (primeira coluna) e com comportamento anormal (segunda coluna) no dataset PETS2009.*

4.1.2 UMN *dataset*

O *dataset* UMN (UMN, 2006) contém vídeos em três locações distintas, todos os vídeos neste *dataset* apresentam anormalidades globais, onde todas as pessoas correm ao mesmo tempo, às vezes na mesma direção e outras vezes em distintas direções.aa

A Figura 23 apresenta exemplos de *frames* dos cenários deste *dataset*. Os vídeos possuem uma resolução de 320x256 pixels, dois deles estão em formato RGB e um em tons de cinza.

Os vídeos deste *dataset* proporcionam anormalidades na Situação 1.

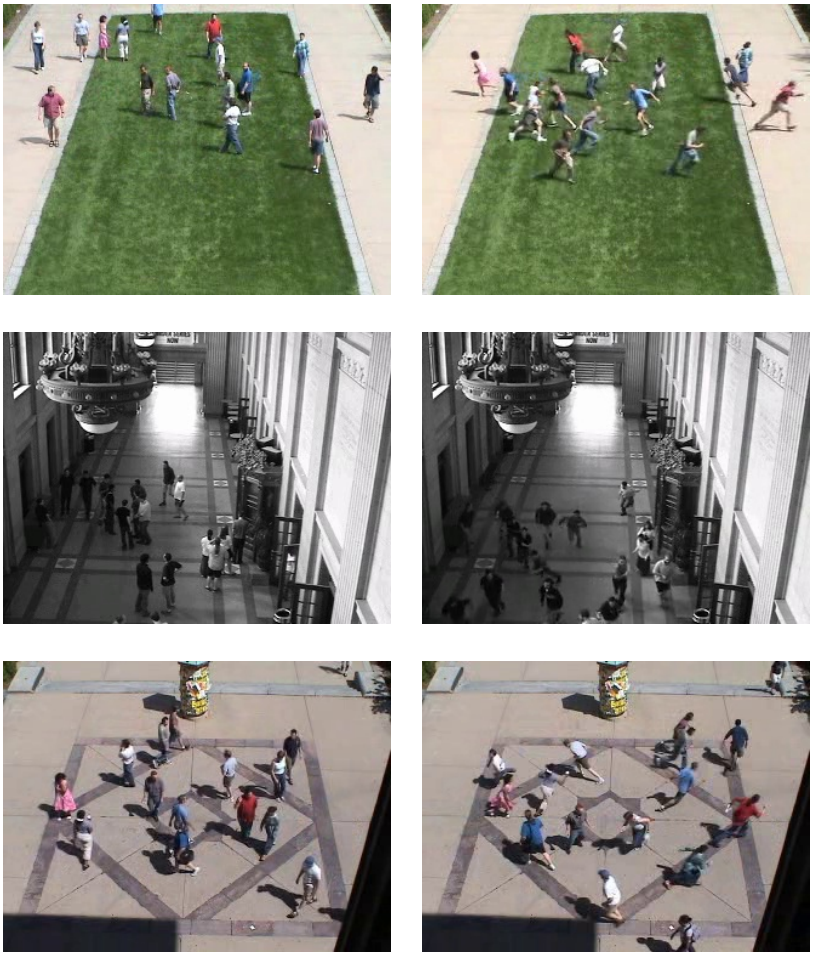


Figura 23 – *Frames de exemplo com comportamento normal (primeira coluna) e com comportamento anormal (segunda coluna) no dataset UMN.*

4.1.3 UCSD *dataset*

O *dataset* UCSD (UCSD, 2013) está composto de um total de 98 vídeos em tons de cinza obtidos em duas locações distintas. A primeira locação, chamada de *UCSDped1*, contem 34 vídeos de treinamento (apresentando unicamente comportamento normal) e 36 vídeos de teste (apresentando algum tipo de anormalidade). A segunda locação, chamada de *UCSDped2*, está composta de 16 vídeos de treinamento e 12 vídeos de teste. A Figura 24 apresenta *frames* de exemplo de cenas com comportamento normal (primeira linha) e comportamento anormal (segunda e terceira linha) tanto do *UCSDped1* (primeira coluna) quanto do *UCSDped2* (segunda coluna).

Os *frames* que compõem os vídeos em *UCSDped1* possuem uma resolução de 238x158 pixels, enquanto os *frames* em *UCSDped2* possuem uma resolução de 360x240 pixels.

Este *dataset* disponibiliza a informação de *ground truth*, ou seja, informa os *frames* dos vídeos de teste onde, segundo o critério do autor do *dataset*, acontecem anormalidades

Figura 24 – *Frames de exemplo de cenas com comportamento normal (primeira linha) e comportamento anormal (segunda e terceira linha) tanto do UCSDped1 (primeira coluna) quanto do UCSDped2 (segunda coluna).*

assim como, em alguns casos, inclui também a localização da anormalidade dentro da imagem. Essas informações são usadas para obter o desempenho do algoritmo proposto.

Os vídeos deste *dataset* proporcionam anormalidades nas Situações 2 e 4.

4.1.4 Matthias Clamer *dataset*

O *dataset* Matthias Clamer (MATTHIAS..., 2012) possui quatro vídeos, todos eles em formato RGB e com dimensões 640x360 pixels. Nos vídeos deste *dataset* um grupo de pessoas caminha em uma direção enquanto uma pessoa caminha em direção contrária. Todas as situações foram filmadas de uma posição superior. Este *dataset* não possui *frames* de treinamento, portanto todos os vídeos são de teste. A Figura 25 apresenta *frames* de exemplos deste *dataset*.

Os vídeos deste *dataset* proporcionam anormalidades na Situação 3.

Figura 25 – *Frames de exemplo do dataset Matthias Clamer.*

4.1.5 IEA-AIE 2014

Este *dataset* é composto por um único vídeo pertencente ao *International Computer Vision Competition* (IEAAIE, 2014). A Figura 26 apresenta *frames* de exemplo contendo comportamento normal e anormal. Este *dataset* foi originalmente disponibilizado para testar algoritmos de contagem de pessoas, no entanto, neste trabalho foi utilizado por apresentar exemplos de anormalidade na Situação 3. Os *frames* deste vídeo possuem uma resolução de 320x240 pixels e estão em formato RGB. É importante destacar que, pelo o conhecimento do autor deste trabalho, é a primeira vez que o vídeo IEA-AIE 2014 é utilizado para avaliar este tipo de algoritmos.

(a) (b)

Figura 26 – *Frames de exemplo do vídeo IEA-AIE 2014. (a) comportamento normal. (b) persona caminhando em direção contraria à multidão: comportamento anormal.*

Os vídeos deste *dataset* proporcionam anormalidades na Situação 3.

A Tabela 4 resume os *datasets* utilizados, a resolução original dos *frames*, a resolução após serem redimensionados e se foi necessária ou não a conversão das imagens de cor a tons de cinza.

Tabela 4 – *Resumo dos datasets utilizados.*

Dataset	Número de vídeos utilizados	Resolução original	Resolução após o pré-processamento	Conversão RGB a tons de cinza
PETS 2009	3	768 x 576	300 x 225	sim
UMN	3	320 x 240	300 x 225	sim
UCSDped1	37	238 x 158	300 x 200	não
UCSDped2	12	360 x 240	300 x 200	não
Matthias Clamer	4	640 x 360	300 x 169	sim
IEA-AIE 2014	1	320 x 240	300 x 225	sim

4.2 Metodologia de avaliação

Os resultados da detecção de comportamento anormal obtidos a partir dos algoritmos propostos neste trabalho foram avaliados e comparados com outros trabalhos correlatos. Dita comparação foi realizada utilizando métricas de desempenho padrão utilizadas na maioria desses trabalhos.

A maioria dos trabalhos na literatura utilizam duas abordagens para medição do desempenho. A primeira é conhecida como *frame-level* e visa avaliar se um *frame* é classificado como contendo ou não anormalidades. A segunda abordagem de avaliação é chamada de *pixel-level* e nela é avaliada a classificação de pixels individuais como sendo normais ou anormais. A primeira abordagem foca-se na identificação da anormalidade enquanto a segunda foca-se na sua localização no *frame*. A Figura 27(a) apresenta um exemplo da abordagem *frame-level* e a Figura 27(b) apresenta um exemplo da abordagem *pixel-level*.

(a) (b)

Figura 27 – *Exemplos de classificação por (a) frame-level, (b) pixel-level.*

A primeira métrica de desempenho a ser usada visa informar graficamente a classificação de *frames* como normais ou contendo alguma anormalidade. Esta métrica permite ter uma visão global do desempenho do classificador e comparar os resultados tanto com o *ground-truth* quanto com outros trabalhos correlatos. A Figura 28 apresenta um exemplo desta métrica onde são usadas barras horizontais representando a longitude total do vídeo. Regiões em cor verde representam *frames* classificados como contendo só comportamento normal; as regiões em cor vermelho representam *frames* classificados como contendo algum tipo de anormalidade. Os números indicam o primeiro *frame* no qual se inicia um conjunto de *frames* classificados como anormais após um período de normalidade.

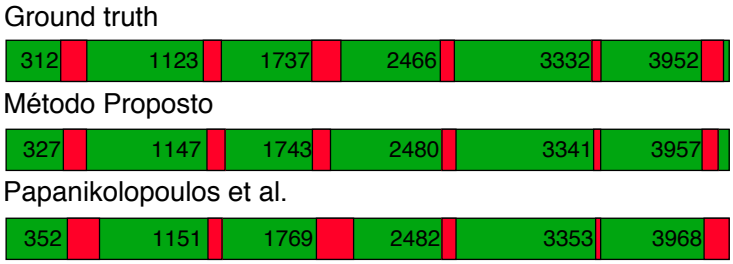


Figura 28 – *Exemplo de métrica de desempenho utilizando barras. Regiões em verde identificam frames classificados como normais. Regiões em vermelho identificam frames classificados como contendo alguma anormalidade. Os números representam o primeiro frame onde se inicia a anormalidade.*

A segunda métrica a ser usada neste trabalho é chamada **curva ROC** (do inglês *Receiver Operating Characteristic*). A curva ROC é uma representação gráfica que ilustra o desempenho de um classificador binário enquanto é variado o parâmetro de discriminação. No eixo horizontal do gráfico são posicionados os valores chamados de razão de Positivos Verdadeiros (RPV), e no eixo vertical são posicionados os valores chamados de razão de Positivos Falsos (RPF). O valor RPV é obtido como a fração entre os Positivos Verdadeiros (PV) e os Positivos Totais (P) e também é chamado de Sensibilidade. O valor RPF é obtido como a fração entre os Positivos Falsos (PF) e os Negativos Totais (N), também é conhecido como 1-Especificidade. É possível incluir várias curvas ROC no mesmo gráfico para comparar o desempenho com outros algoritmos. A Figura 29 apresenta um exemplo de curva ROC para três algoritmos diferentes. Quanto mais se aproximar a curva ao ponto (0,1), ou seja, ao canto superior esquerdo, melhor desempenho apresentará o algoritmo.

Da curva ROC é possível obter mais duas métricas de desempenho comumente utilizadas. A primeira é a área sub a curva ROC conhecida como **AUC** (do inglês *Area Under the Curve*) e a segunda é conhecida como o **EER** (do inglês *Equal Error Rate*). A AUC é igual à probabilidade de um classificador atribuir um valor maior a uma amostra positiva escolhida randomicamente do que a uma mostra negativa, também escolhida randomicamente. **Quanto maior for a AUC melhor**. O EER é o ponto onde a razão de Falsos Positivos é igual à razão de Falso Negativos. No gráfico do ROC o EER é valor no eixo horizontal do ponto onde ocorre a interseção entre a curva ROC e a reta que

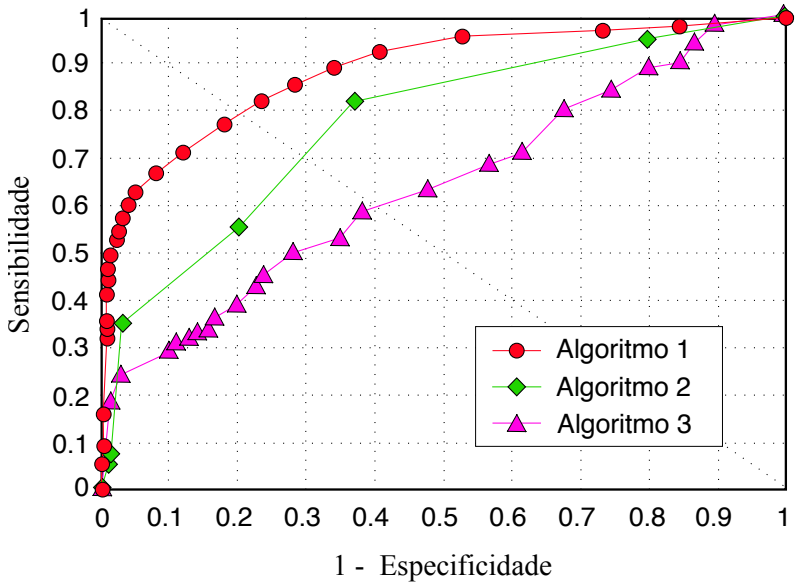


Figura 29 – *Exemplo de curva ROC para três algoritmos. O Algoritmo 2 apresenta melhor desempenho que o Algoritmo 3, no entanto, apresenta pior desempenho que o Algoritmo 1*

une os pontos (0,1) e (1,0). **Quanto menor for o EER melhor**. Em geral a AUC e o EER são apresentados em porcentagem. Por exemplo, na Figura 29 o Algoritmo 1 apresenta um EER de aproximadamente 20%, o Algoritmo 2 apresenta um EER de 30% aproximadamente e o Algoritmo 3 apresenta um EER de 40% aproximadamente.

Dependendo do algoritmo e das informações disponíveis nos trabalhos da literatura, será usada uma ou várias das métricas apresentadas anteriormente.

Na sequência serão apresentadas as implementações e os resultados dos algoritmos de detecção de comportamento anormal.

4.3 Caso 1: Implementação e resultados

4.3.1 Considerações de implementação

Inicialmente são definidos os tamanhos da fila longa (L) e da fila curta (C) (ver Figura 11 do Capítulo 3). Esses parâmetros são ajustáveis e podem influir no desempenho do algoritmo. Em primeiro lugar deseja-se que o algoritmo responda rapidamente às mudanças de magnitude do fluxo óptico na cena e, levando em conta que, segundo a proposta inicial do algoritmo, são necessários $C/2$ *frames* com comportamento anormal para detectar a anormalidade, é necessário que o valor de C seja baixo, portanto o valor $C = 15$foi utilizado nos testes. Por outro lado, o valor L deve ser muito maior do que C, portanto o valor utilizado nos testes é $L = 35$.

Após serem obtidos os primeiros $C + L = 50$ *frames*, e os valores das filas curta e longa serem armazenados como descrito na Seção 3.2, a cada novo *frame* são calculados os valores de média (μ) e desvio padrão (σ) dos valores contidos na fila longa $\boldsymbol{f}_l$, assim como a mediana ($\tilde{f}_c$) dos valores contidos na fila curta $\boldsymbol{f}_c$. A seguir são obtidos os limites de pertinência do modelo gaussiano: $\mu - 3 * \sigma$ e $\mu + 3 * \sigma$. A Figura 30 apresenta os valores dos limites de pertinência, o valor da mediana na fila curta ($\tilde{f}_c$) e o momento onde é detectada a situação de anormalidade para o vídeo apresentado na primeira linha da Figura 22.

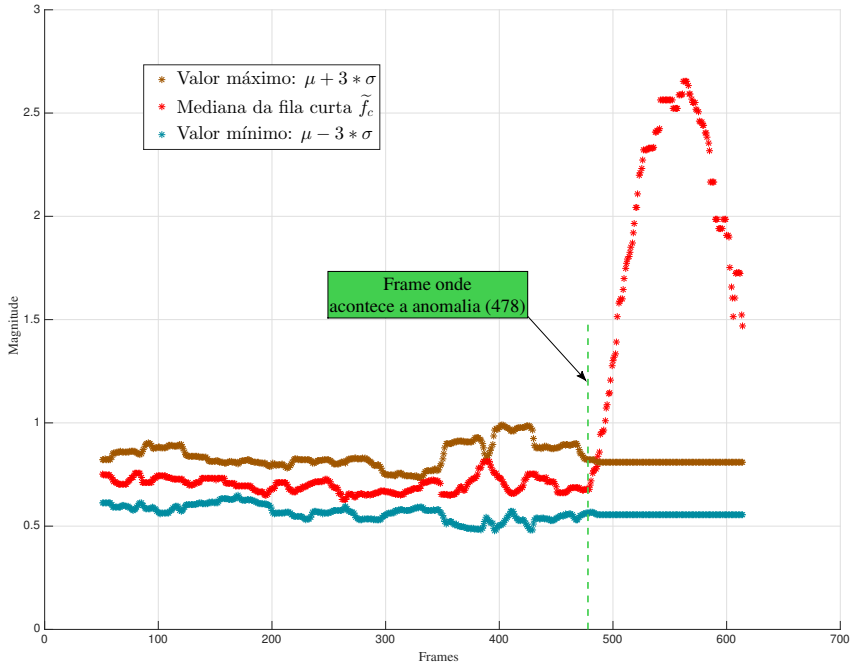


Figura 30 – *Variação do intervalo de pertinência do modelo Gaussiano da fila longa: valores mínimos $\mu-3*\sigma$ e máximos $\mu+3*\sigma$. Em vermelho a mediana dos valores da fila curta.*

A partir do gráfico apresentado na Figura 30 é possível perceber uma variação grande do intervalo de pertinência. Essa variação é causada pelos poucos dados armazenados na fila longa. Uma possível abordagem para estabilizar os limites é armazenar um maior número de valores. A desvantagem dessa abordagem é a necessidade de calcular a média e o desvio padrão de um conjunto de dados cada vez maior e, se a sequência de imagens for longa, aumentaria consideravelmente o tempo de processamento entre *frames*. Portanto, neste trabalho foi utilizado o algoritmo de Welford (KNUTH, 1997) o qual obtém a média e o desvio padrão de dados adicionados continuamente. Usando esse algoritmo é possível combinar cada novo dado com os dados anteriores sem precisar armazená-los. Como resultado, a fila longa não é mais necessária. Assim, são definidos os valores de média total (μ_k) e desvio padrão total (σ_k) no instante k calculados segundo as seguintes equações,

$$\mu_k = \mu_{k-1} + (\overline{m}_k - \mu_{k-1})/k \tag{4.1}$$

$$S_k = S_{k-1} + (\overline{m}_k - \mu_{k-1}) * (\overline{m}_k - \mu_k) \tag{4.2}$$

onde $\overline{m}$ é o último elemento da fila curta, e μ_k é a média das amostras até o instante k. O desvio padrão é dado por

$$\sigma_k = S_k/(k-1) \tag{4.3}$$

A Figura 31 apresenta a nova versão do algoritmo. O valor da mediana da fila curta ($\tilde{f}_c$) é comparada com o intervalo de pertinência obtido a partir dos valores μ_k e σ_k sendo: $\mu_k - 3 * \sigma_k$ e $\mu_k + 3 * \sigma_k$.

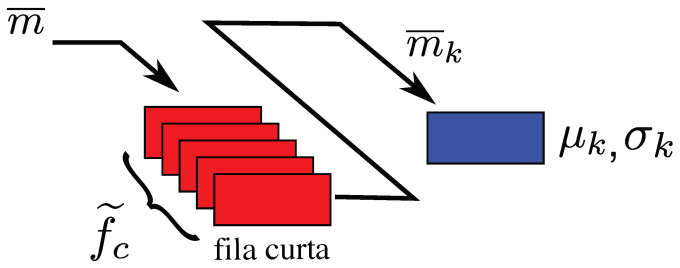


Figura 31 – *Fila curta e obtenção dos valores μ_k e σ_k*

A nova abordagem traz duas vantagens: a primeira é uma redução na variação dos limites de pertinência da distribuição Gaussiana, o que produz maior estabilidade e menor taxa de erros por falsos positivos. A segunda vantagem é a redução de carga computacional no momento de calcular a média e o desvio padrão total, já que não é mais necessário calcular esses valores para uma lista, e o tempo de cálculo é o mesmo sem importar o número de *frames* do vídeo.

A Figura 32 apresenta os novos intervalos de pertinência para o mesmo vídeo utilizado na Figura 30. Pode-se observar que os limites de pertinência são mais estáveis, porém, o algoritmo requer um número maior de *frames* para detectar o comportamento anormal: no exemplo da Figura 30 o comportamento anormal é detectado no *frame* 487, enquanto utilizando a média e o desvio padrão μ_k e σ_k a detecção é realizada no *frame* 489, requerendo 2 *frames* a mais. No entanto a estabilidade e menor sensibilidade ao ruído compensa a necessidade de maior número de *frames*.

Na Figura 33 é apresentada uma comparação entre os limites mínimo e máximo para o mesmo vídeo de teste onde é evidenciada a vantagem de utilizar a média e o desvio padrão μ_k e σ_k: na Figura 33(a) os limites são calculados somente para os dados atuais contidos na fila longa ($\boldsymbol{f}_l$), aqui aparecem três falsos positivos. Na Figura 33(b) os limites são calculados utilizando a nova técnica, obtendo μ_k e σ_k, portanto leva em conta o histórico completo dos valores em $\boldsymbol{f}_l$, evitando os falsos positivos e obtendo um modelo Gaussiano mais estável.

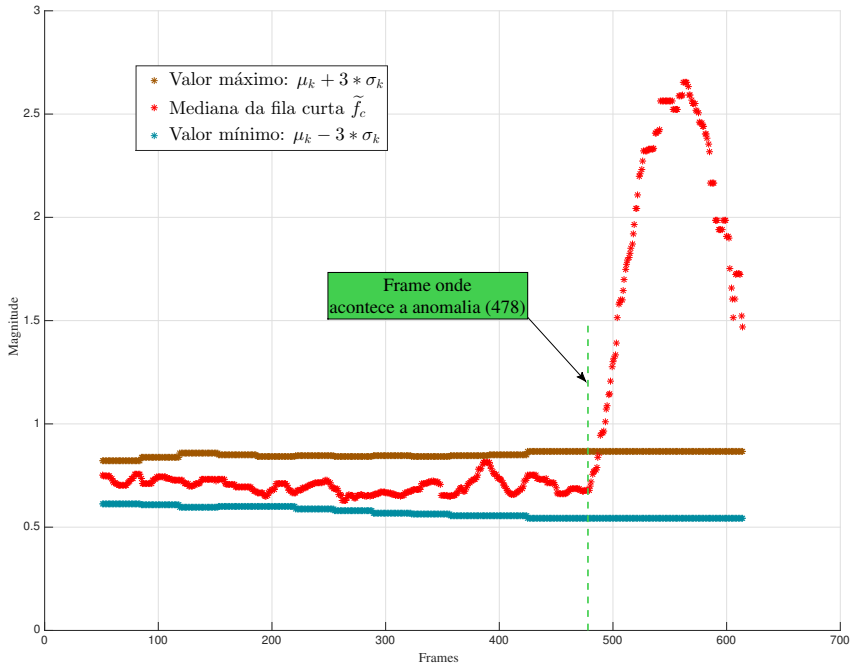


Figura 32 – *Intervalos de pertinência do modelo Gaussiano, limite inferior:* $\mu_k - 3 * \sigma_k$*; limite superior:* $\mu_k + 3 * \sigma_k$ *e mediana da fila curta* $\tilde{f}_c$.

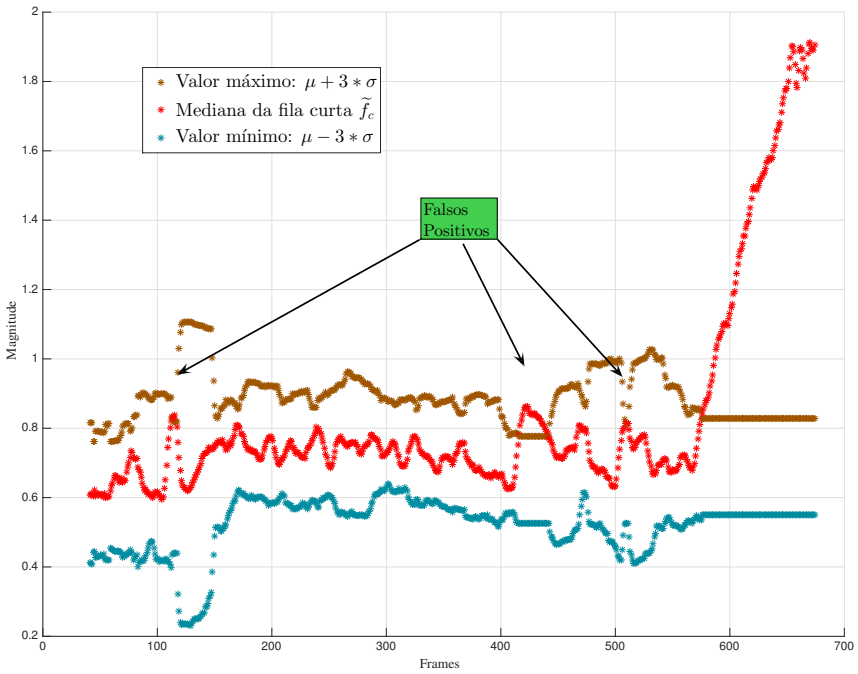
4.3.2 Resultados

O método proposto foi testado nos *datasets* PETS 2009 e UMN. A Figura 34 apresenta os resultados obtidos para os vídeos de teste no *dataset* PETS 2009 utilizando o método gráfico das barras horizontais. A primeira barra mostra a classificação do *frame* esperada (conhecido como *ground truth*), enquanto a segunda barra mostra a classificação obtida pelo algoritmo proposto.

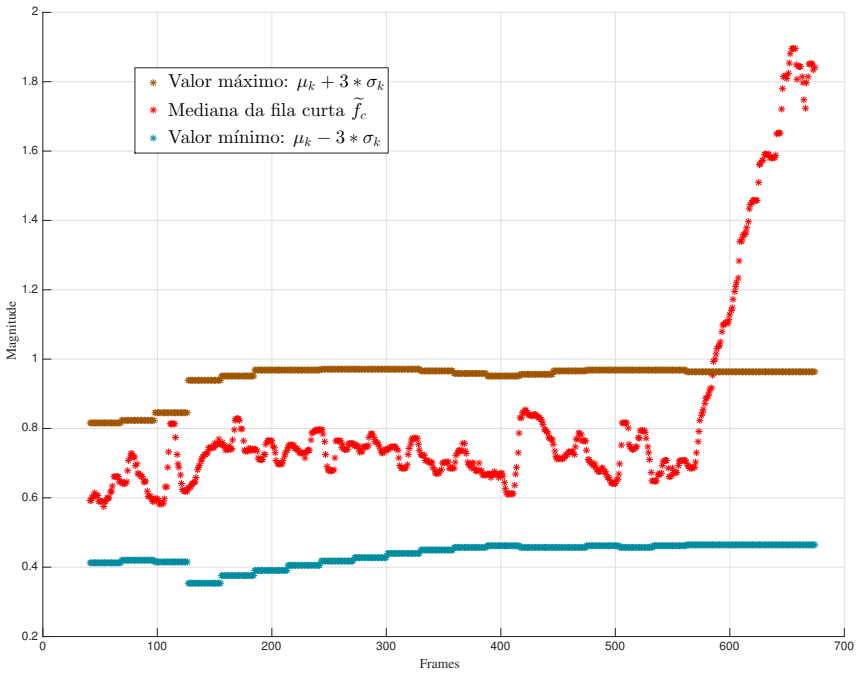
As Figuras 35, 36 e 37 apresentam o resultado obtido para o *dataset* UMN. De novo, a primeira barra mostra o *ground thruth*, a segunda barra o resultado obtido pelo método proposto e as restantes barras são os resultados reportados em (FRADI; DUGELAY, 2014), (ALMEIDA; JUNG, 2013) e (CHEN; HUANG, 2011).

4.3.3 Avaliação dos resultados

Conforme os resultados obtidos para esta situação, pode-se concluir que o algoritmo consegue classificar corretamente a maioria dos *frames* como sendo normais ou contendo a anormalidade de aumento repentino de velocidade. A abordagem apresentada é simples e continuamente atualizável por não se basear em um modelo previamente obtido, precisando poucos recursos de memória. O intervalo de pertinência permite variar a sensibilidade do algoritmo: um intervalo de pertinência menor detectará pequenas mudanças na velocidade; enquanto um intervalo de pertinência maior requer um aumento de velocidade maior para ser classificada como anormalidade.



(a)



(b)

Figura 33 – *Comparação entre os limites mínimo e máximo para o mesmo vídeo de teste (a) sem histórico e (b) com histórico.*

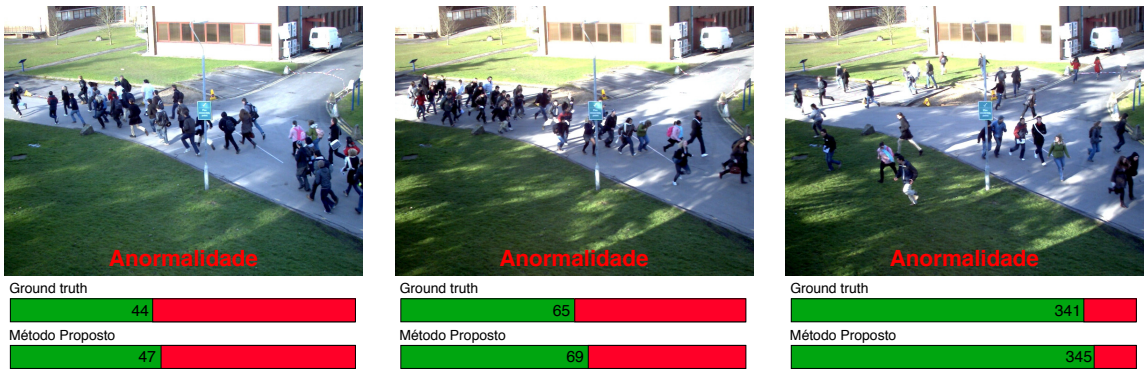


Figura 34 – *Resultados do algoritmo proposto para o dataset PETS2009.*

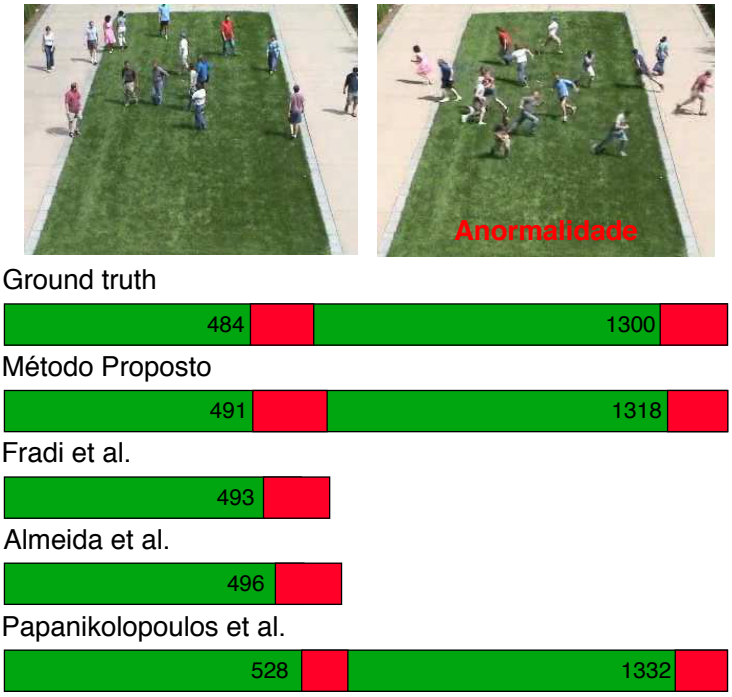


Figura 35 – *Comparação de resultado obtidos para a cena 1 do dataset UMN usando a métrica de barras horizontais.*

4.4 Caso 2: Implementação e resultados

4.4.1 Considerações de implementação

Neste módulo visa-se identificar anormalidades nos padrões de presença, velocidade e direção de movimento de forma local. Ele foi dividido em três partes detalhadas nas Secções 3.3.1, 3.3.2 e 3.3.3. A primeira parte (Seção 3.3.1) visa identificar e localizar movimento em áreas na imagem onde pouco ou nenhum movimento foi detectado anteriormente. A segunda parte (Seção 3.3.2) visa identificar anormalidades na direção de movimentação e finalmente, a terceira parte (Seção 3.3.3) visa identificar e localizar anormalidades onde a magnitude de deslocamento de uma ou mais pessoas é muito diferente

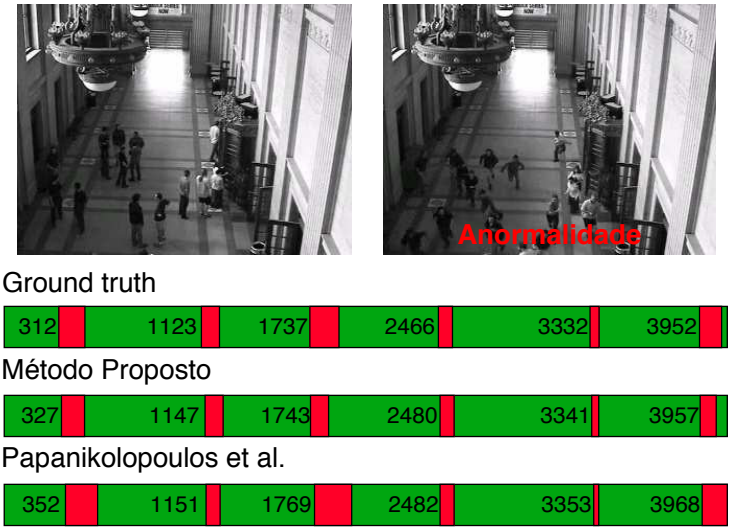


Figura 36 – *Comparação de resultado obtidos para a cena 2 do dataset UMN.*

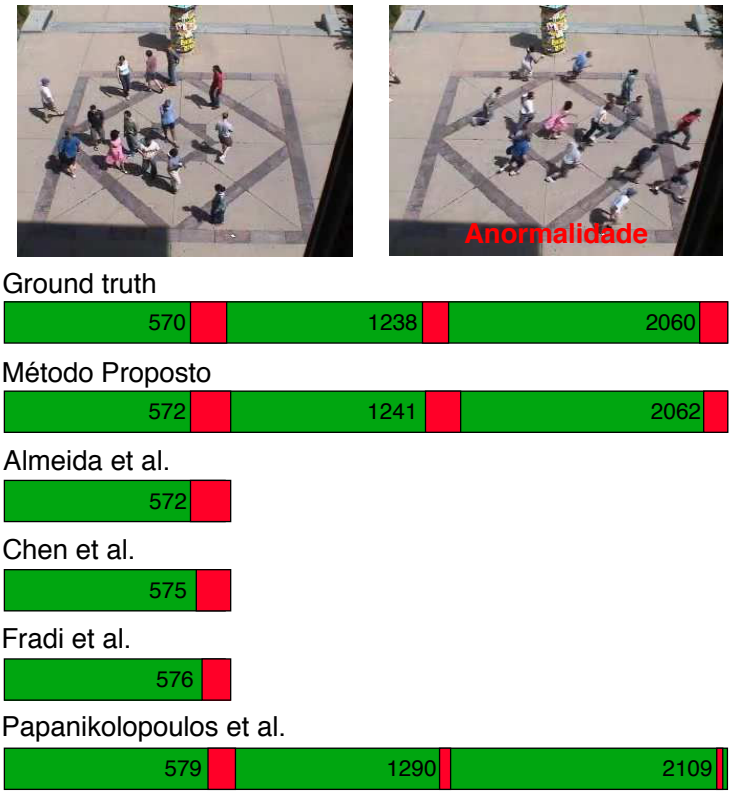


Figura 37 – *Comparação de resultado obtidos para a cena 3 do dataset UMN.*

dos padrões presentes no modelo.

Inicialmente as três partes compartilham a mesma entrada: a matriz de magnitude do fluxo óptico filtrada $\widetilde{\boldsymbol{M}}$ e a matriz de direções do fluxo óptico filtrada $\widetilde{\boldsymbol{D}}$. Essas duas matrizes são divididas em regiões retangulares de tamanho fixo denotadas como R_i, as quais, por sua vez, são divididas em sub-regiões quadradas de tamanho fixo denotadas como Sr_i^j, onde o índice i representa o número da região e o índice j representa o número da sub-região dentro da região i (ver Figura 14 do Capítulo 3).

As regiões R_i e as sub-regiões Sr_i^j são implementadas como sendo regiões de interesse (ROI pelas siglas em inglês) dentro das matrizes $\widetilde{\boldsymbol{M}}$ e $\widetilde{\boldsymbol{D}}$. Portanto, cada região e sub-região é definida como um quarteto de números organizados em um vetor linha, onde o primeiro e o segundo número indicam as coordenadas x e y do canto superior esquerdo da região enquanto o terceiro e o quarto número indicam a largura e altura da mesma.

Para determinar as coordenadas de cada região e sub-região o primeiro passo é definir a largura e altura de cada sub-região. Nesta implementação foram definidas sub-regiões quadradas, portanto, a largura é igual à altura. Definida a largura de cada sub-região deve-se estabelecer o número de sub-regiões que comporão cada região estabelecendo os valores F_{sr} e C_{sr}. Com os valores F_{sr} , C_{sr} e a largura de cada sub-região é possível calcular tanto largura como a altura de cada região. Obtidos esses valores e junto com as dimensões da imagem, é possível obter o número de regiões horizontais (F_r) e o número de regiões verticais (C_r) que comporão a grade de regiões. Caso as dimensões da grade de regiões não sejam múltiplo do tamanho das matrizes $\widetilde{\boldsymbol{M}}$ e $\widetilde{\boldsymbol{D}}$ então serão acrescentadas tantas linhas e colunas de zeros como sejam necessárias.

Visando obter uma localização mais precisa da anormalidade dentro da imagem assim como obter modelos redundantes que ajudem a diminuir os falsos positivos, nos testes realizados neste módulo foi empregada uma grade de regiões onde existia uma sobreposição das mesmas. Como consequência o número de regiões aumentou assim como o número total de sub-regiões. A Tabela 5 resume os valores adotados nos testes dos algoritmos neste módulo assim como o número de regiões e sub-regiões obtidos com e sem sobreposição.

Cada uma das partes que compõem este módulo foi avaliada de forma separada permitindo uma visualização da desempenho de cada algoritmo.

Cada algoritmo neste módulo retorna uma imagem binária contendo uns nos pixels que apresentam anormalidade: o algoritmo de detecção de anormalidade por presença retorna a imagem $\boldsymbol{\mathcal{P}}$ o de detecção de anormalidade por direção retorna $\boldsymbol{\mathcal{F}}$ e anormalidades por velocidade retorna $\boldsymbol{\mathcal{B}}$.

Tabela 5 – *Parâmetros de teste dos algoritmos do módulo 2.*

Parâmetro	Valor sem sobreposição	Valor com sobreposição
Largura/altura de cada sub-região em pixels	4	4
F_{sr}	5	5
C_{sr}	5	5
Largura de cada região em pixels	20	20
Altura de cada região em pixels	20	20
F_r	10	19
C_r	15	29
Número total de regiões	150	551
Número total de sub-regiões	3750	13775

4.4.2 Resultados

Os resultados obtidos neste módulo serão divididos em três secções: a Seção 4.4.2.1 apresentará os resultados da detecção de anormalidade por presença, a Seção 4.4.2.2 apresentará os resultados para anormalidades por direção e a Seção 4.4.2.3 para anormalidade por velocidade.

4.4.2.1 Detecção de anormalidade por presença

Para obter o modelo de presença, ou seja, estabelecer os valores do histograma $\boldsymbol{H}_p$, são utilizados todos os *frames* dos vídeos de treinamento. No caso do *dataset* UCSDped1 foram utilizados no total 6499 *frames* de treinamento, enquanto, no caso do *dataset* UCSDped2, foram utilizados 2550 *frames*.

A Figura 38 apresenta o histograma de presença para o *dataset* UCSDped1. A Figura 39 apresenta as zonas com maior frequência de presença (em amarelo) e as de menor ou nenhuma frequência de presença (em azul) para o mesmo cenário.

O *dataset* UCSDped1 contém três vídeos de teste onde acontecem anormalidades por presença: *Test009*, *Test011* e *Test031*. A Figura 40 apresenta um *frame* de exemplo de cada um dos vídeos testados com a detecção e localização da anormalidade. Neste exemplo foram utilizados valores de $P_{max} = 500$ e $\xi_p = 0,12$ o que resultou em um valor $P_{min} = 60$. Portanto se alguma sub-região apresenta informação de magnitude de fluxo óptico e seu valor de frequência no histograma $\boldsymbol{H}_p$ for menor do 60, então é detectada a anormalidade nessa sub-região.

Para avaliar a desempenho deste algoritmo é utilizada a curva ROC onde o parâmetro de discriminação variado é ξ_p.

A Figura 41 apresenta a curva ROC para cada uma das três cenas de teste variando

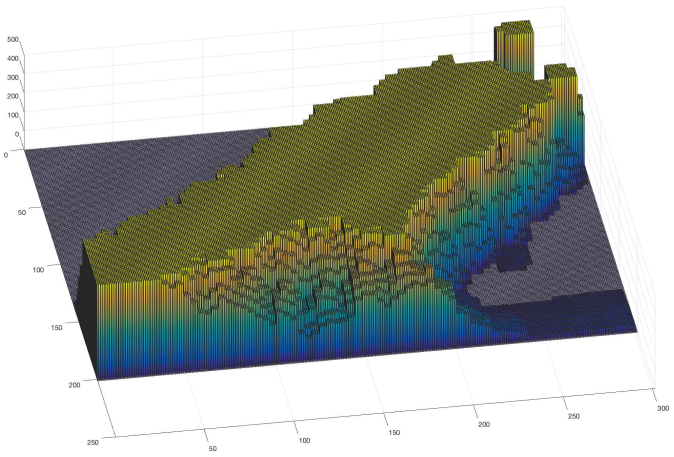


Figura 38 – *Histograma de presença para o dataset UCSDped1.*

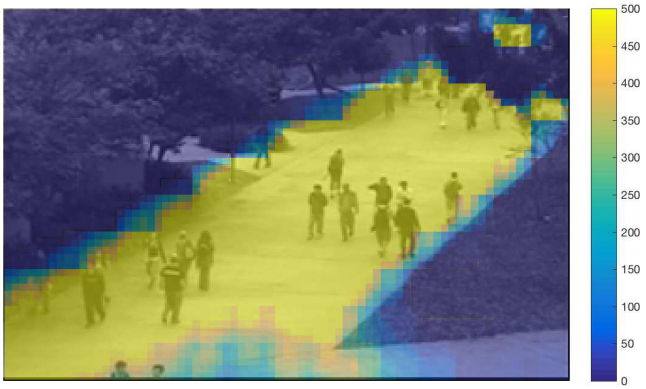


Figura 39 – *Mapa de presença para o dataset USDCped1.*

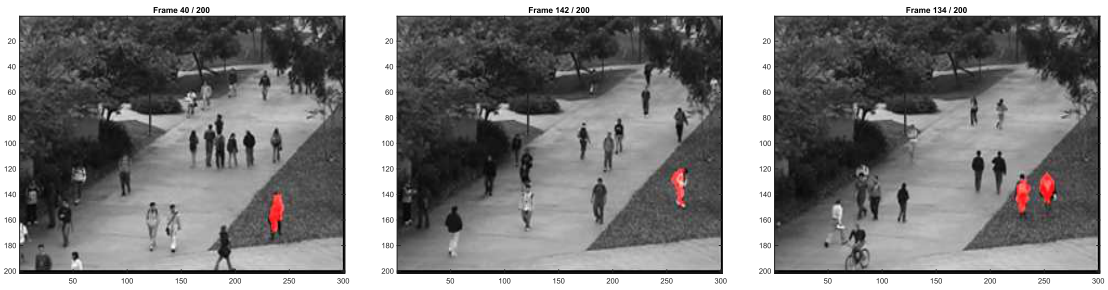


Figura 40 – *Exemplos de detecção de anormalidade por presença no dataset UCSDped1.*

o parâmetro ξ_p entre 0 e 1 aumentando a razão de 0,05.

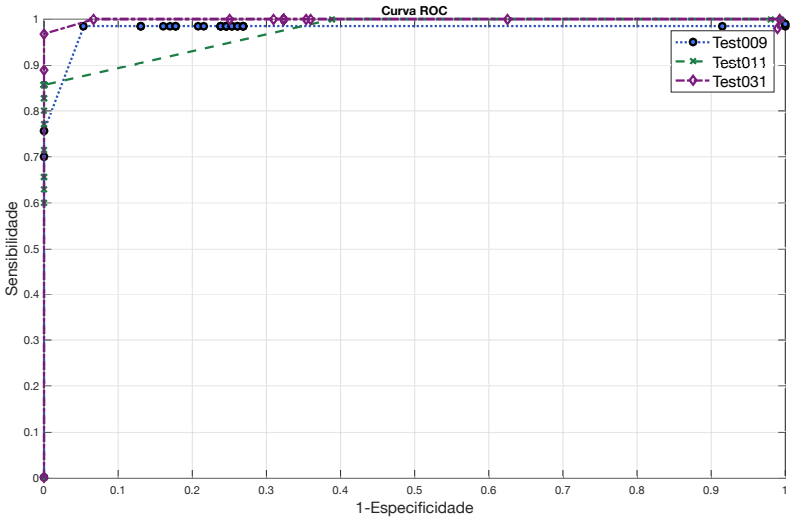


Figura 41 – *Curva ROC para os três vídeos de teste no dataset UCSDped1.*

A Figura 42 apresenta o histograma de presença para o *dataset* UCSDped2 e a Figura 43 as zonas com maior frequência de presença para o mesmo *dataset*. Das duas imagens é possível concluir que, na parte inferior da cena, existe muito pouco ou nenhum deslocamento de pessoas nos vídeos de treinamento o que leva ao algoritmo a detectar pessoas ou objetos se deslocando nessa região como anormalidade.

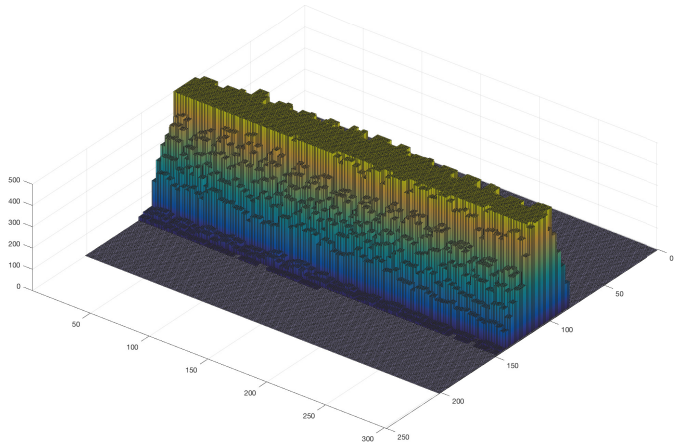


Figura 42 – *Histograma de presença para o dataset UCSDped2.*

A Figura 44 apresenta exemplos de anormalidade por presença nos vídeos de teste *Test003*, *Test006* e *Test007* do dataset UCSDped2 e a Figura 45 apresenta as curvas ROC para cada um dos vídeos.

4.4.2.2 Detecção de anormalidade por direção

O *dataset* UCSD não está focado na detecção de anormalidades por direção, portanto unicamente foi identificado um vídeo (*Test011*) em UCSDpeds1 onde existe esse tipo

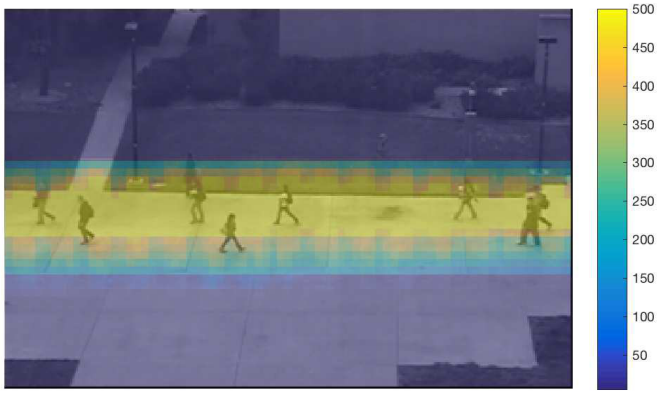


Figura 43 – *Mapa de presença para o dataset USDCped2.*

Figura 44 – *Exemplos de detecção de anormalidade para o dataset UCSDped2.*

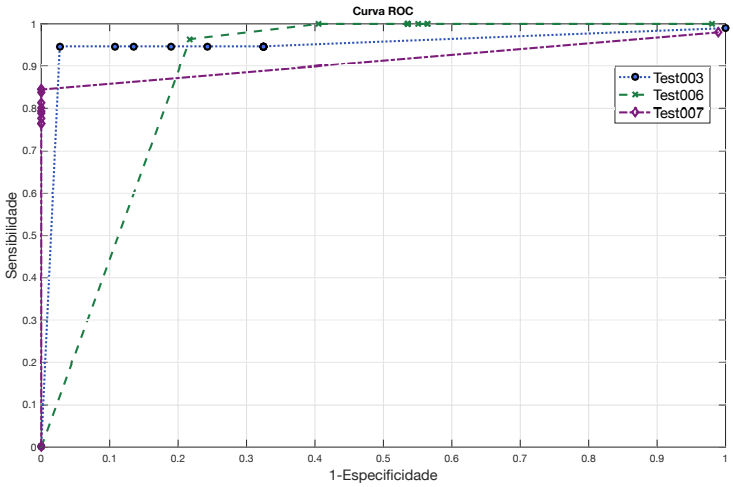
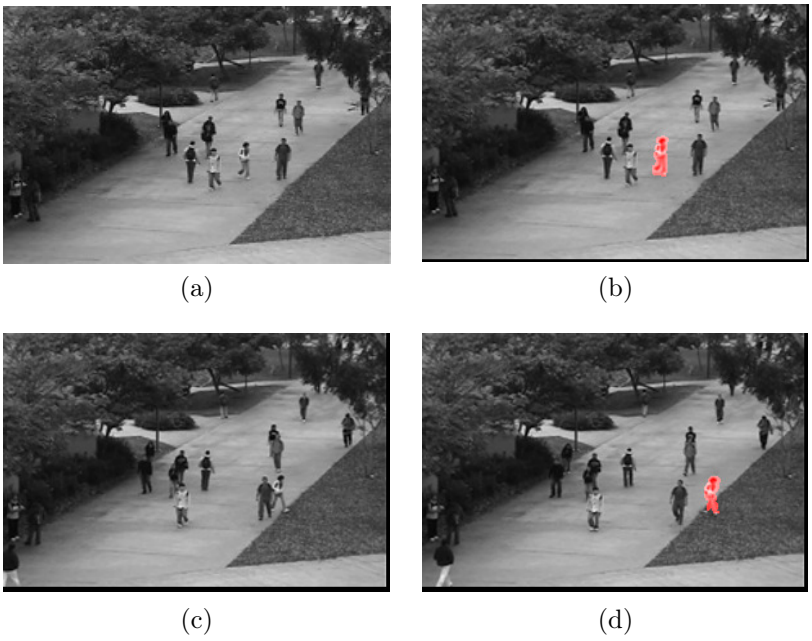


Figura 45 – *Curva ROC para três vídeos de teste no dataset UCSDped2.*

de anormalidade. A Figura 46(a) apresenta um *frame* de entrada ao algoritmo detalhado na Seção 3.3.2. Na Figura 46(b) é apresentada a identificação da anormalidade por parte do algoritmo proposto. Na Figura 46((c) é mostrado um exemplo de Falso Negativo, onde o algoritmo não detecta a anormalidade devido ao fato da pessoa com a anormalidade passar por trás de outra pessoa, o que produz oclusão e portanto não é possível obter o fluxo óptico com a anormalidade. Na Figura 46((d) é identificada de novo a anormalidade. As saídas apresentadas na Figura 46 foram obtidas utilizando $T_D = 0,001$.



(a) (b)

(c) (d)

Figura 46 – *Exemplo de detecção de anormalidade por direção. (a) entrada do algoritmo com anormalidade, (b) saída do algoritmo, (c) falso negativo causado por oclusão e (d) detecção da anormalidade.*

O autor deste *dataset* fornece o *ground-truth* para este vídeo o que permite obter a curva ROC. O *ground-truth* fornecido contém informação de anormalidade ao nível de *frame*, portanto, é comparada a classificação por parte do algoritmo aqui proposto de cada *frame* como contendo anormalidade ou não. A curva ROC variando o parâmetro de discriminação T_D sendo $T_D = f(x) = 2^x$ onde x varia de 0 até 0,14 incrementado em 0,003 é apresentada na Figura 47.

4.4.2.3 Detecção de anormalidade por velocidade

A maior parte dos vídeos de testes no *dataset* UCSD apresentam anormalidades associadas a pessoas em bicicleta, ou carros passando pela travessia de pedestres. Embora essas anormalidades possam ser associadas a objetos (bicicletas, skates ou carros) que aparecem pela primeira vez nos vídeos de teste, podem ser identificados neste módulo dado que, geralmente, esses elementos se deslocam com uma velocidade superior à velocidade de deslocamento das pessoas nos vídeos de treinamento.

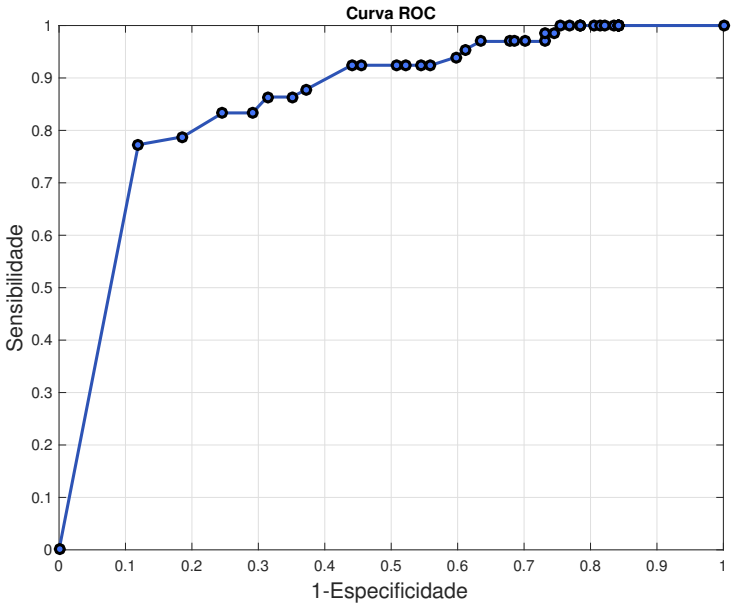


Figura 47 – *Curva ROC para teste de anormalidade de direção.*

A Figura 48 apresenta exemplos de identificação de anormalidades por magnitude obtidos utilizando o algoritmo descrito na Seção 3.3.3.

A Figura 49 apresenta a curva ROC de todos os vídeos de teste levando em conta só a análise por velocidade. A curva foi obtida ao variar o parâmetro T_i (a distância mínima entre o novo mapa de magnitudes e os *templates* de cada região).

A Figura 50 apresenta exemplos de resultados no *dataset* UCSDped2 e a Figura 51 apresenta a curva ROC para todos os vídeos de teste neste *dataset*.

Finalmente, a Figura 52 apresenta exemplos de saída do módulo completo, onde anormalidades por presença são destacada com a cor amarelo, anormalidades por direção são destacadas com a cor verde e aquelas classificadas como anormalidade por velocidade são destacada em vermelho.

4.4.3 Avaliação dos resultados

Dos resultados obtidos neste módulo pode-se concluir que o algoritmo apresenta um bom desempenho na detecção de anormalidades por presença, direção e velocidade de movimentação. Na Figura 52 pode-se atestar o benefício de utilizar módulos especializados para identificar os diferentes tipos de anormalidade. Até o conhecimento do autor, nenhum trabalho na literatura obtém informação sobre o tipo de anormalidade. Informação que pode ser utilizada na tomada de decisões.

A abordagem por histogramas oferece a possibilidade de adicionar novas informações ao modelo, assim como escolher a sensibilidade do algoritmo às mudanças a partir dos parâmetros ξ_p, T_D e ξ_m. Também foi constatada que a detecção de anormalidades por

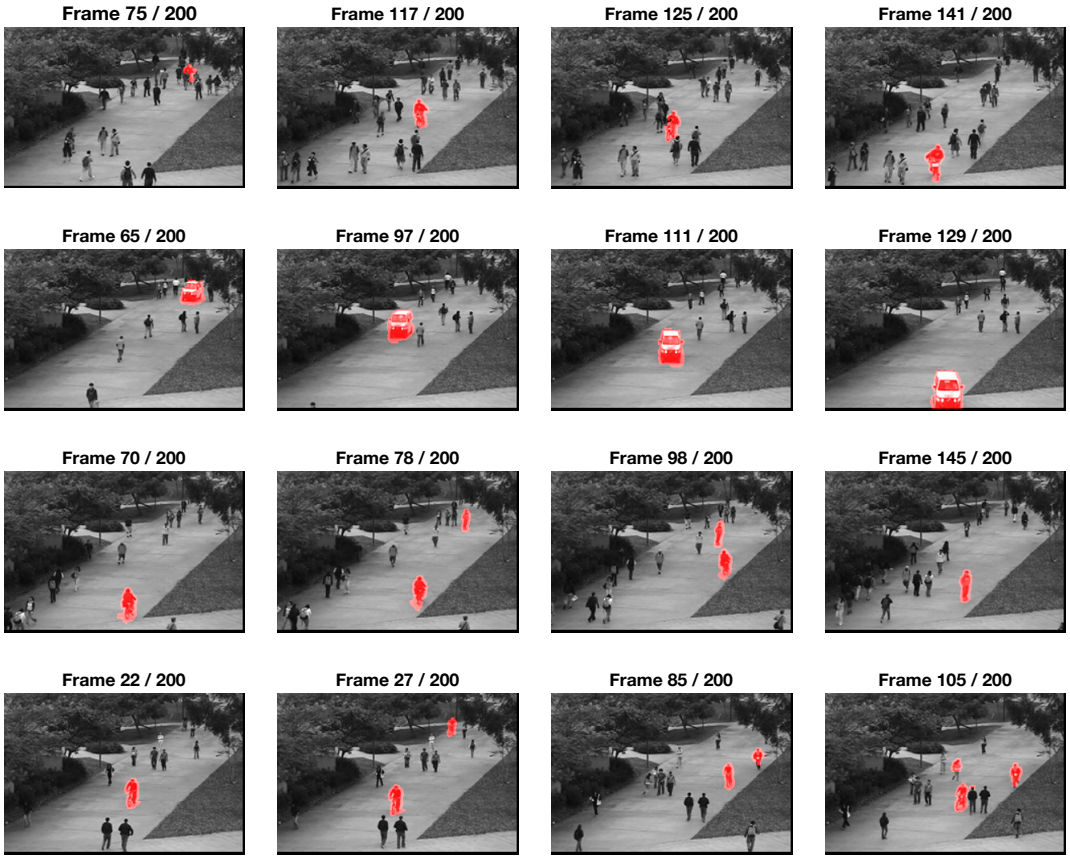


Figura 48 – *Exemplos de detecção de anormalidade por velocidade no dataset UCSDped1.*

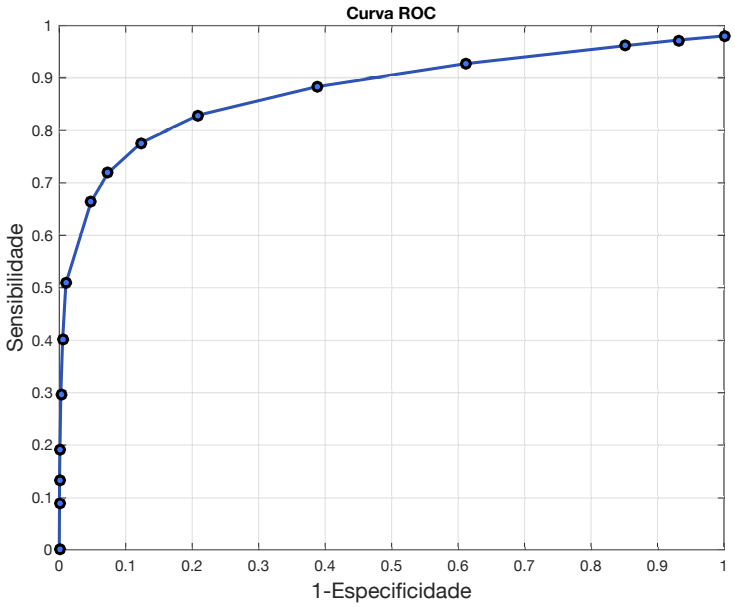


Figura 49 – *Curva ROC para todos os vídeos de teste do dataset UCSDped1.*

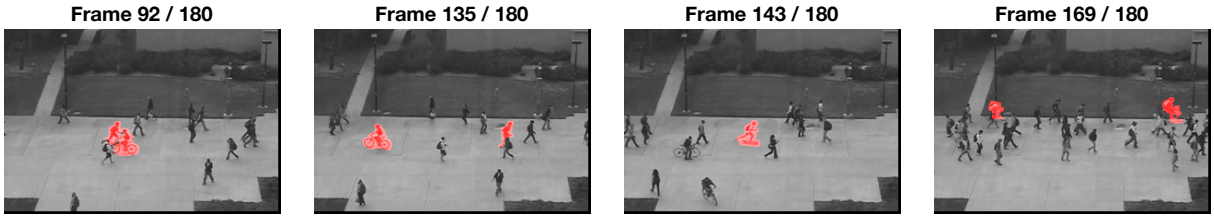


Figura 50 – *Exemplos de detecção de anormalidade por velocidade no dataset UCSDped2.*

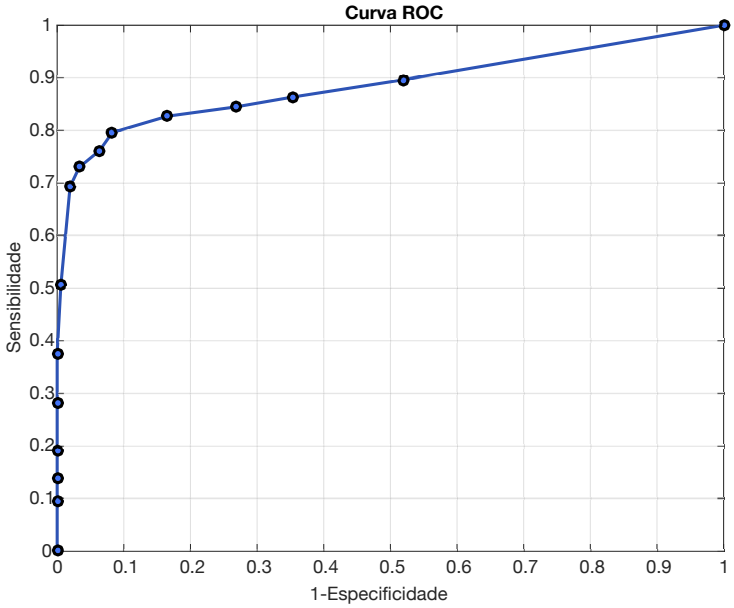


Figura 51 – *Curva ROC para todos os vídeos de teste do dataset UCSDped2.*

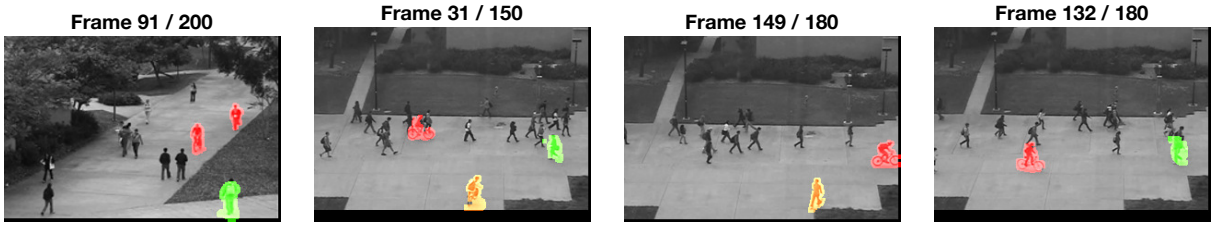


Figura 52 – *Exemplos de saída combinada. Em amarelo anormalidade por presença, em verde anormalidade por direção e em vermelho anormalidade por velocidade.*

velocidade é a parte que mais tempo de processamento consome. Isso devido a que esse algoritmo foi planejado para ser o mais genérico possível, permitindo processar cenas onde existam vários padrões de velocidade na mesma região, como por exemplo, calçadas onde carros e pessoas ocupam o mesmo espaço com padrões de velocidade diferentes. Portanto, aumenta a complexidade do algoritmo e, por consequência, seu tempo de processamento.

4.5 Caso 3: Implementação e resultados

Nesta seção serão apresentados os resultados obtidos ao implementar o algoritmo proposto na Seção 3.4 para a detecção de anormalidades quando uma ou poucas pessoas deslocam-se em direção contrária ao resto da multidão. Para avaliar este algoritmo foram usados os vídeos do *dataset* Matthias Clamer, assim como o vídeo IEA-AIE 2014 e um vídeo do *dataset* PETS 2009.

4.5.1 Considerações de implementação

Nesta implementação dois parâmetros são estabelecidos pelo usuário: uma porcentagem da área total da imagem que deve ter um componente conexo para considerá-lo na análise de direção, denotada por ϕ; e o valor de quantização de ângulos $\Delta\theta$. No caso da porcentagem da área o valor adotado foi $\phi = 0,01$. Portanto, deve-se cumprir que a área do componente conexo seja superior a 0,01 vezes a área total da imagem para prosseguir sua análise. O valor adotado para o parâmetro de quantização de ângulos foi de $\Delta\theta = 10$. Os resultados obtidos com esses parâmetros serão apresentados na seção a seguir.

4.5.2 Resultados

A Figura 53 apresenta *frames* de exemplo com as anormalidades obtidas utilizando o algoritmo apresentado na Seção 3.4.

A Tabela 6 apresenta os valores da razão de Positivos Verdadeiros e da razão de Positivos Falsos para cada vídeo de teste usado neste módulo. Os vídeos do *dataset* Matthias Clamer estão numerados como 879-42, 879-43, 879-44 e 879-45.

4.5.3 Avaliação dos resultados

Dos resultados apresentados anteriormente pode-se concluir que o algoritmo apresenta uma boa razão de Positivos Verdadeiros (RPV), e baixos valores na razão de Positivos Falsos (RPF). Igualmente, da Figura 53, conclui-se que o algoritmo apresenta uma boa localização na imagem da pessoa (ou pessoas) que está gerando a anormalidade.

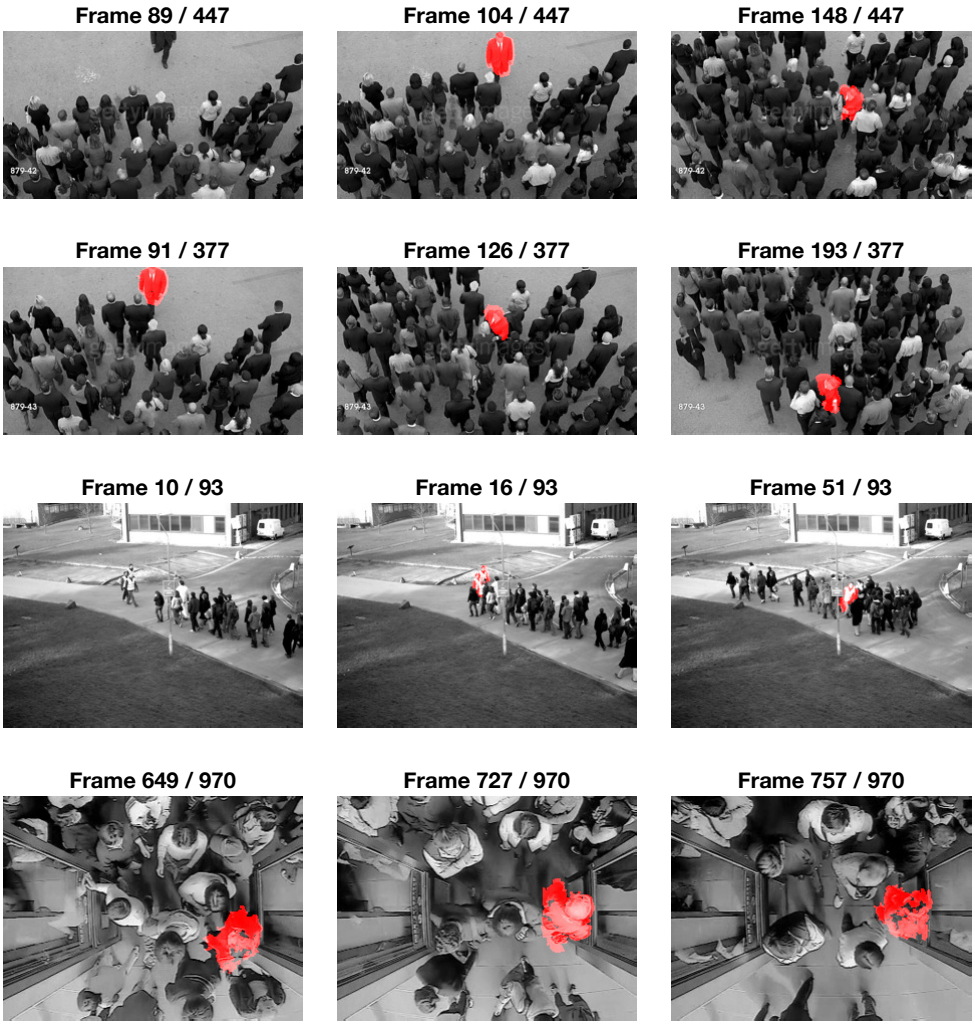


Figura 53 – *Exemplos de anormalidades associadas a pessoas caminhando em direção contraria à multidão.*

Tabela 6 – *Razão de Positivos Verdadeiros (RPV) e Razão de Positivos Falsos (RPF) para os vídeos de teste do módulo 3.*

Video	RPV (%)	RPF (%)
879-42	90,1	0
879-43	97,9	0,86
879-44	94,7	0,5
879-45	48	0
Pets01	35,5	9
Pets02	70,3	9,6
IEA-AIE 2014	42,2	0

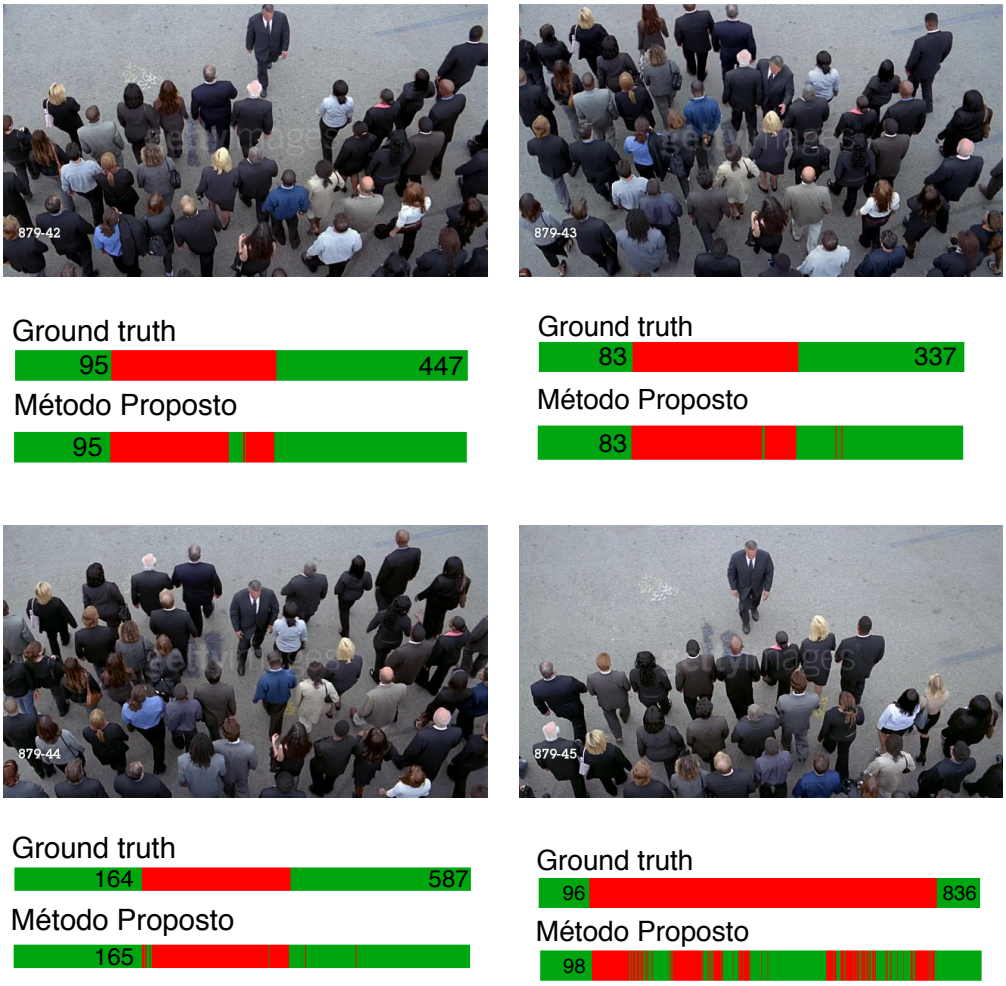


Figura 54 – *Resultados de detecção de anormalidade para o dataset Matthias Clamer.*

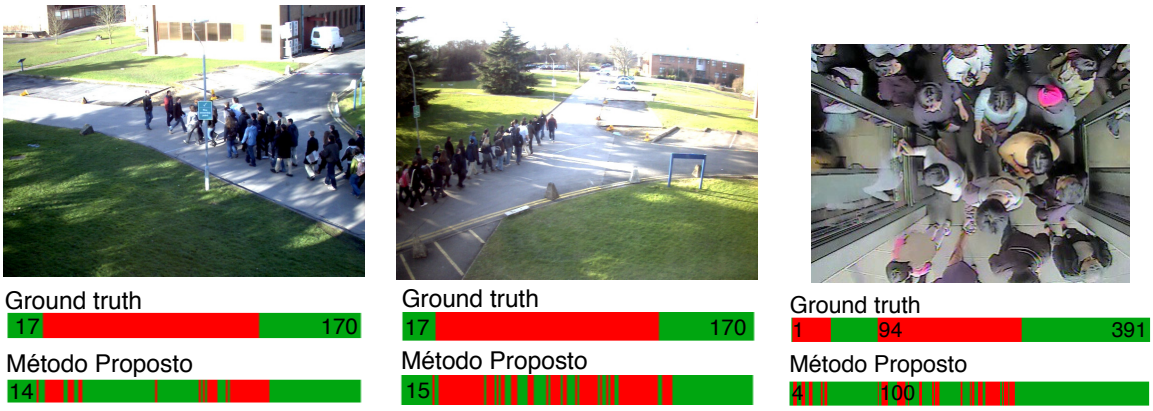


Figura 55 – *Resultados de detecção de anormalidade para os datasets PETS 2009 e IEA-AIE 2014.*

Da métrica de avaliação por barras e da Tabela 6 percebe-se que no último vídeo do *dataset* Matthias Clamer (879-45), assim como no vídeo Pets01 e IEA-AIE 2014 existe uma baixa RPV. Isso acontece porque, no caso dos vídeos 879-45 e Pets01, a densidade de pessoas na multidão é muito alta o que impede à pessoa em contramão se deslocar. Sem deslocamento não é gerado fluxo óptico, portanto não é possível identificar a anormalidade. No caso do vídeo IEA-AIE 2014 as pessoas em contramão detêm-se em várias ocasiões. E, mais uma vez, sem movimento não é gerada a informação do fluxo óptico. Uma possível solução a essa limitante é analisar a dinâmica de movimento da multidão procurando regiões com movimento restrito causado pela pessoa em contramão. Outra possibilidade é, uma vez detectada a anormalidade, manter a alarme enquanto a pessoa está detida até ela se movimentar de novo. Caso nenhuma anormalidade seja detectada em um período de tempo estabelecido, o alarme é desligado.

4.6 Caso 4: Implementação e resultados

Nesta seção serão apresentados os resultados obtidos ao implementar o algoritmo proposto na Seção 3.5 para a detecção de anormalidades por aparência. Neste módulo não é utilizada a informação do fluxo óptico, unicamente será necessária a máscara do *foreground* obtida pelo algoritmo de subtração de fundo.

4.6.1 Considerações de implementação

O primeiro parâmetro a ser levado em conta neste módulo é o tamanho da região R_i. Foram testados vários valores de R_i podendo ser regiões quadradas ou retangulares, com ou sem sobreposição. Os resultados obtidos apresentam uma alta sensibilidade a este parâmetro, devido a que pequenas mudanças nele retornam resultados de classificação diferentes. O segundo parâmetro a ser levado em conta é a distância máxima T_a. O valor desse parâmetro dependerá da métrica de distância a ser utilizada, uma vez que métricas diferentes têm faixas de valores diferentes. Nesta implementação foram testados vários valores de $\Delta\theta$. Em geral, na literatura, $\Delta\theta$ possui valores entre 20 e 45 graus. Nos resultados aqui apresentados foi fixado o valor de $\Delta\theta = 30^{\text{o}}$ o que significa que o número de barras no histograma de orientações do gradiente é $\eta = 12$. Por fim, o parâmetro de classificação T_ϕ também dependerá da métrica de distância adotada. Na sequência serão apresentados os resultados obtidos utilizando vários valores para os parâmetros de distância máxima e T_ϕ.

4.6.2 Resultados

A Figura 56 apresenta alguns exemplos de resultados obtidos para detecção de anormalidade por aparência.

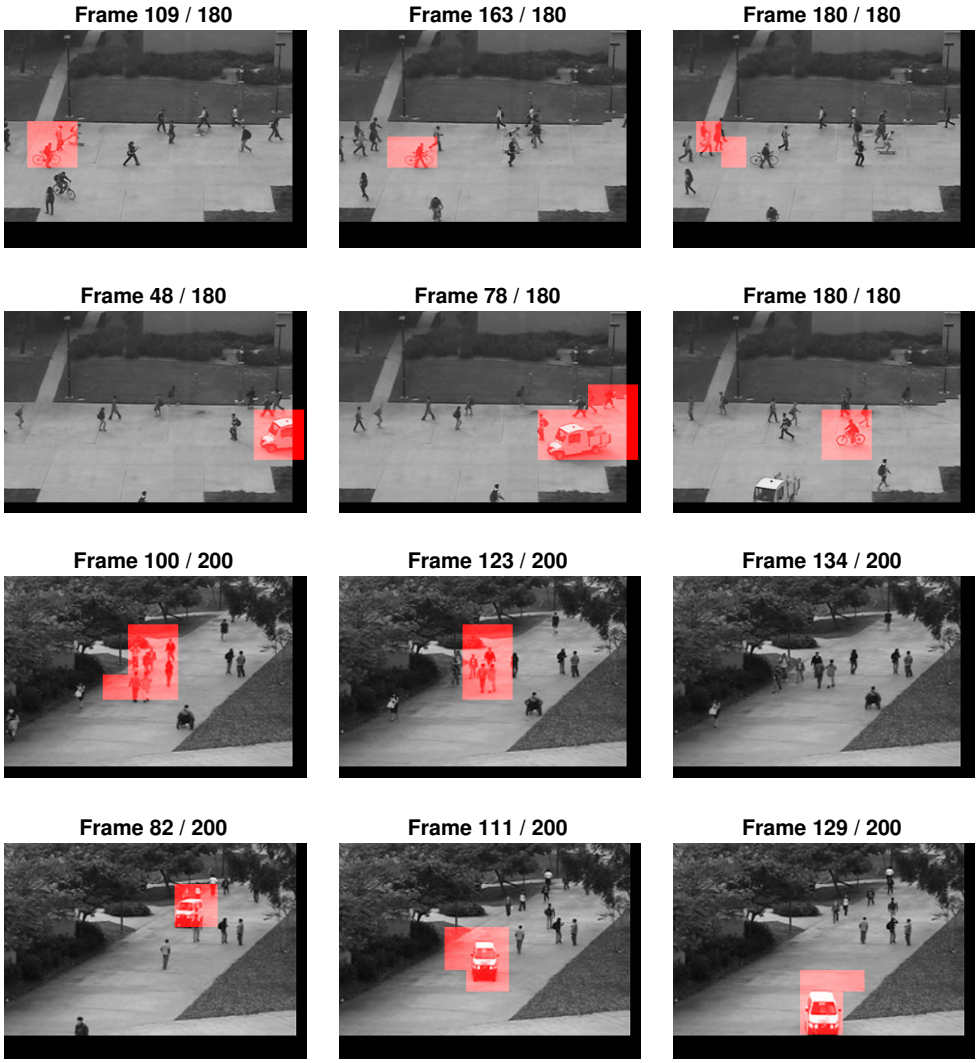


Figura 56 – *Exemplos de anormalidade por apareça no dataset UCSD.*

A Tabela 7 apresenta a Razão de Positivos Verdadeiros (RPV) e a Razão de Positivos Falsos (RPF) para duas regiões de tamanho fixo de 44x52 pixels utilizando várias métricas. A primeira região de teste R_1 contém tanto *frames* com anormalidades quanto *frames* sem anormalidades. A segunda região de teste R_2 contém unicamente *frames* sem anormalidade.

4.6.3 Avaliação dos resultados

Da Figura 56 pode-se concluir que o algoritmo classifica corretamente objetos grandes como carros e, em alguns casos, bicicletas; mas falha na classificação de objetos

Tabela 7 – *Razão de Positivos Verdadeiros (RPV) e Razão de Positivos Falsos (RPF) para os vídeos de teste do módulo 3.*

			R_1		R_2	
Métrica	T_a	T_ϕ	RPV (%)	RPF (%)	RPV (%)	RPF (%)
Euclidiana	50	0,002	85,48	10,16	0	11,67
Correlação	0,4	1,5	64,51	2,54	0	11,67
Cosseno	0,05	2	53,22	16,94	0	10
Emd	100	4,5e-5	88,70	11,01	0	5,56
L1	60	1,5e-4	96,77	17,8	0	12,22

menores como a cadeira de rodas. Também percebe-se que o algoritmo gera falsos positivos em regiões onde existe uma densidade alta de pessoas. O baixo desempenho apresentado neste módulo pode ser justificada pelo grande número de possíveis formas que pode apresentar o gradiente na imagem de cinza, assim como a ausência de uma análise temporal do gradiente. Portanto, se conclui que o histograma das orientações do gradiente usado neste trabalho calculado a cada *frame* não consegue capturar toda a dinâmica espaço-temporal necessária para discriminar objetos baseados na sua aparência. Cabe mencionar que foram feitos testes utilizando a técnica de HOG tradicional utilizada em vários trabalhos correlatos, mas o desempenho obtido utilizando unicamente o HOG foi pior que o obtido usando a técnica detalhada neste trabalho. Como resultado é necessária uma nova abordagem ou melhora no algoritmo atual, para obter um desempenho similar ao obtido nos anteriores módulos.

4.7 Combinador

Nesta seção será descrita a proposta do combinador utilizada no presente trabalho. Embora existam muitas possibilidades de implementação do combinador foi priorizada a simplicidade e fácil inclusão de futuros módulos. A Figura 57 apresenta a proposta implementada neste trabalho. As entradas do combinador são as saídas dos módulos que compõem o sistema. Cada módulo pode gerar uma ou várias imagens binárias com uns nos pixels classificados como tendo alguma anormalidade. Cada uma dessas imagens binárias é multiplicada por um número binário, podendo ser zero ou um, chamado de peso. Em seguida o combinador realiza a operação OR com todas as matrizes de entrada gerando uma única matriz de saída $\boldsymbol{\mathcal{G}}$ contendo todas as anormalidades identificadas na imagem que, empregando os módulos detalhados neste trabalho, é obtida segundo a

seguinte expressão

$$\mathcal{G} = (\omega_1 * \mathcal{H}_1) \vee (\omega_{\mathcal{P}} * \mathcal{P}) \vee (\omega_{\mathcal{F}} * \mathcal{F}) \vee (\omega_{\mathcal{B}} * \mathcal{B}) \vee (\omega_3 * \mathcal{H}_3) \vee (\omega_4 * \mathcal{H}_4) \tag{4.4}$$

onde $\mathcal{H}_1$ é a imagem binária produzida pelo primeiro módulo de detecção de anormalidades, $\mathcal{P}$, $\mathcal{F}$, e $\mathcal{B}$ são produzidas pelo segundo módulo, $\mathcal{H}_3$ pelo terceiro módulo e $\mathcal{H}_4$ pelo quarto módulo.

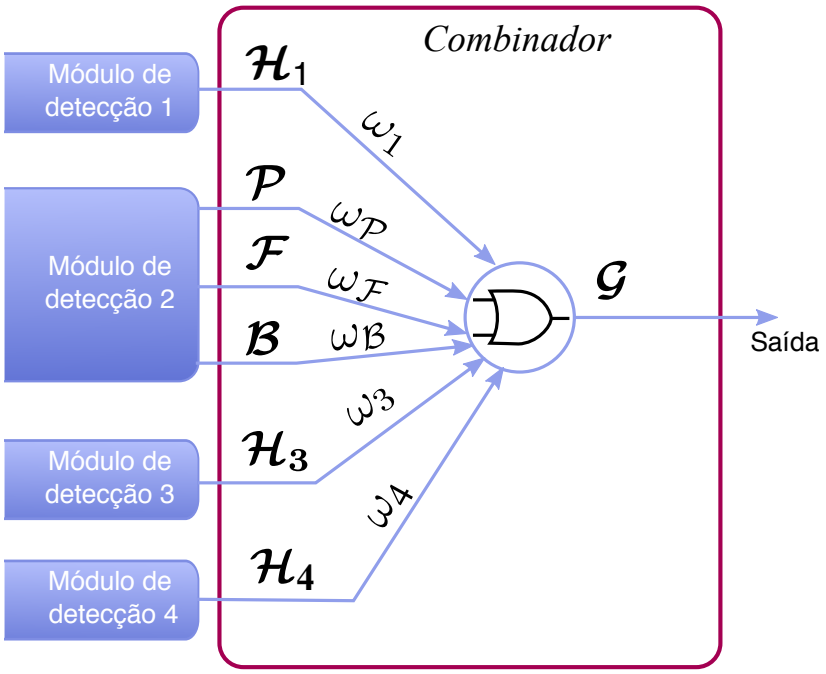


Figura 57 – *Implementação do combinador.*

A implementação adotada permite escolher que tipos de anormalidades desejam ser detectadas pelo sistema em um momento específico. Por exemplo, se o usuário não estiver interessado nas anormalidades por aparência, ele poderia estabelecer o peso correspondente em zero, neste caso $\omega_4 = 0$ enquanto todos os demais pesos são estabelecidos em um. Também seria possível modificar os pesos automaticamente segundo outros critérios, como por exemplo a hora do dia ou o número de pessoas na cena, etc.

O combinador é o último elemento do sistema. Na sequência será apresentada a comparação dos resultados obtidos com o sistema proposto e outros trabalhos correlatos.

4.8 Comparação de resultados com trabalhos correlatos

Nesta seção será apresentada a comparação entre os resultados obtidos pelo sistema proposto e alguns dos trabalhos correlatos detalhados no Capítulo 2.

Até o conhecimento do autor, não existe um único *dataset* que contenha todas as anormalidades detectadas pelos algoritmos descritos neste trabalho, e, levando em conta que a maioria dos trabalhos correlatos reportam o desempenho de seus algoritmos utilizando únicamente o *dataset* UCSD, as métricas de desempenho aqui apresentadas serão obtidas utilizando unicamente esse *dataset*. No entanto, esse *dataset* só apresenta anormalidades de direção, de velocidade e de presença e, portanto, só será necessário o módulo dois para detectá-las. No entanto, as métricas de desempenho aqui apresentadas são obtidas da saída total do sistema, ou seja, após o combinador.

As Figuras 58 e 59 apresentam as curvas ROC utilizando todos os vídeos de teste para os *datasets* UCSDped1 e UCSDped2 respectivamente, comparando o resultado obtido com as curvas reportadas nos trabalhos correlatos. Da mesma forma, as Tabelas 8 e 9 apresentam as métricas AUC e o EER detalhadas na Seção 4.2 para os dois *datasets*. Em ditas tabelas quando um trabalho não reporta uma determinada métrica ela é substituída por -.

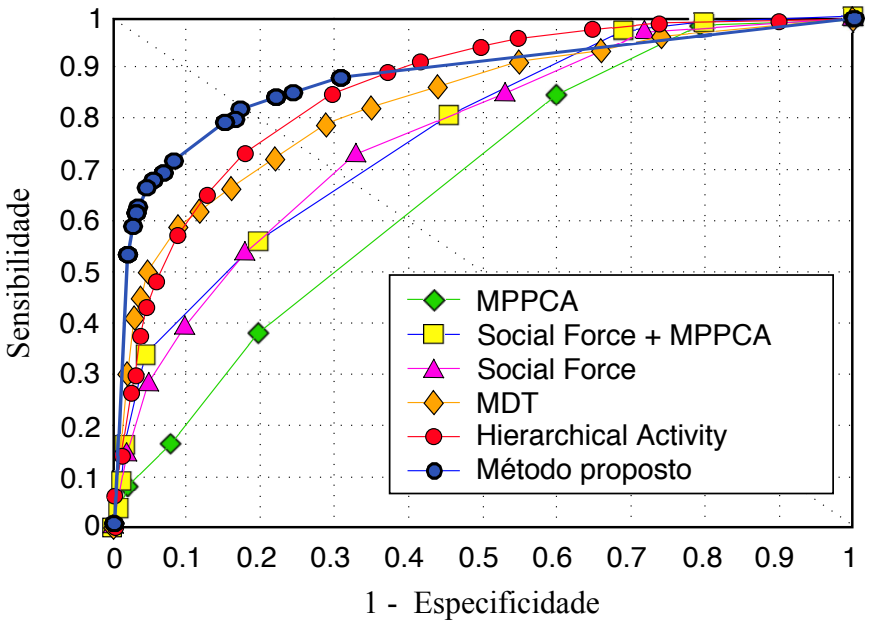


Figura 58 – *Curva ROC para todos os vídeos de teste no dataset UCSDped1.*

Tabela 8 – *Comparação de EER e AUC para o dataset UCSDped1.*

Método	EER	AUC
MPPCA (KIM; GRAUMAN, 2009)	40%	59%
Social Force (SF) (MEHRAN *et al.*, 2009)	31%	67.5%
MPPCA + SF (MAHADEVAN *et al.*, 2010)	32%	68.8%
MDT (MAHADEVAN *et al.*, 2010)	25%	81.8%
Hierarchical Activity (XU *et al.*, 2014)	-	85.4%
Social Attribute (CHEN *et al.*, 2015) 26%	-	
Swarm (KALTSA *et al.*, 2015)	27%	-
Temporal Regularity (HASAN *et al.*, 2016)	27.9%	81%
Deep-cascade (SABOKROU *et al.*, 2017)	9.1%	-
Método proposto	17%	88.4%

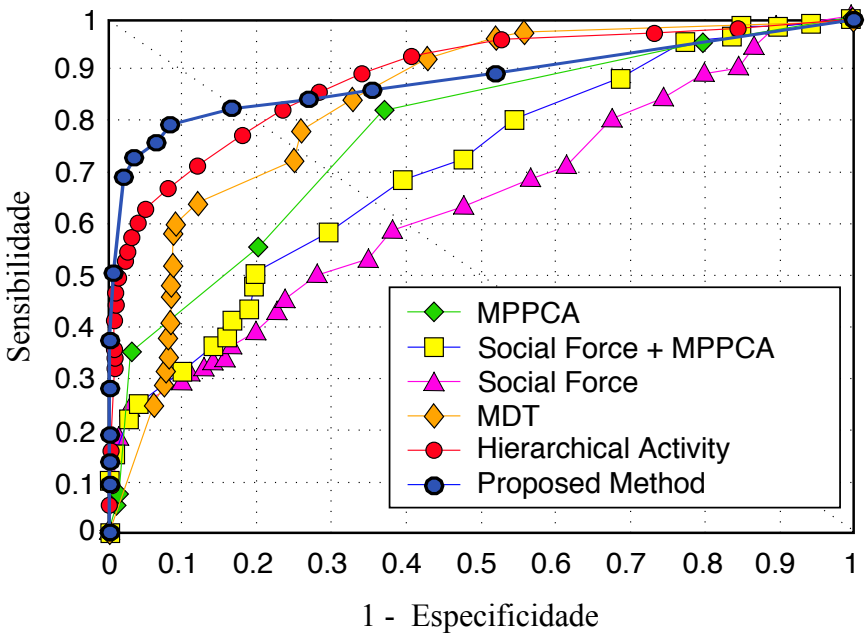


Figura 59 – *Curva ROC para todos os vídeos de teste no dataset UCSDped2.*

Tabela 9 – *Comparação de EER e AUC para o dataset UCSDped2.*

Método	EER	AUC
MPPCA (KIM; GRAUMAN, 2009)	30%	69.3%
Social Force (SF) (MEHRAN *et al.*, 2009)	42%	55.6%
MPPCA + SF (MAHADEVAN *et al.*, 2010)	36%	61.3%
MDT (MAHADEVAN *et al.*, 2010)	25%	82.9%
Hierarchical Activity (XU *et al.*, 2014)	-	88.2%
Social Attribute (CHEN *et al.*, 2015)	-	-
Swarm (KALTSA *et al.*, 2015)	26.9%	-
Temporal Regularity (HASAN *et al.*, 2016)	21.7%	90%
Deep-cascade (SABOKROU *et al.*, 2017)	8.2%	-
Método proposto	17%	88.5%

As Figuras 61 e 60 apresentam exemplos de *frames* com anormalidades destacando sua localização (*pixel-level*) utilizando o sistema proposto comparando-as com as reportadas em outros trabalhos correlatos. Na Figura 61 observa-se que no vídeo 31, *frame* 129, o método proposto é o único que identifica as duas pessoas caminhando sobre a grama, graças à detecção de anormalidades por presença.

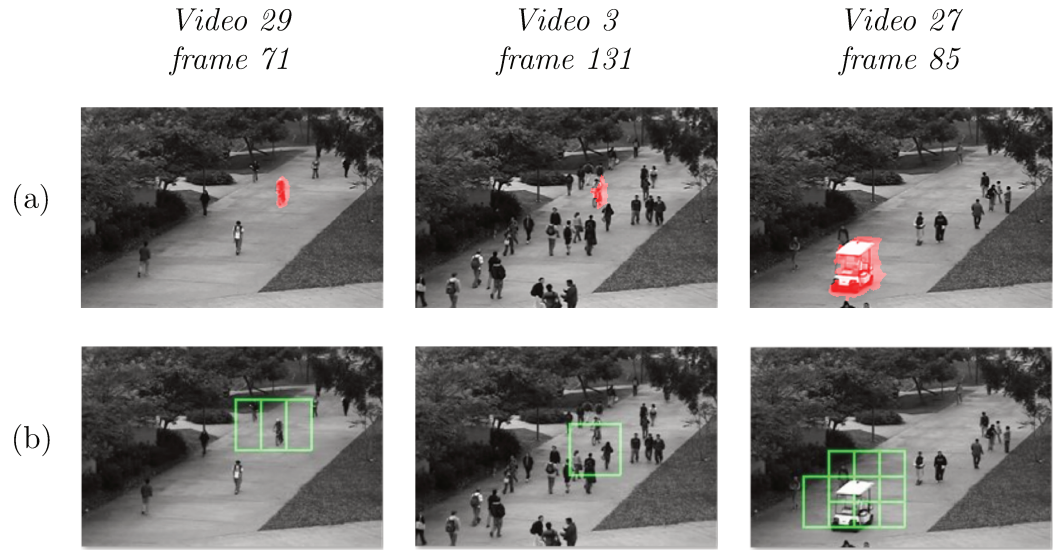


Figura 60 – *Resultados de localização em alguns frames de exemplo no dataset UCSDped1 para (a) sistema proposto e (b) deep-cascade (SABOKROU* et al.*, 2017).*

Das curvas ROC apresentadas nas Figuras 58 e 59, e das Tabelas 8 e 9 é possível concluir que o método proposto neste trabalho obteve o menor valor EER para detecção de

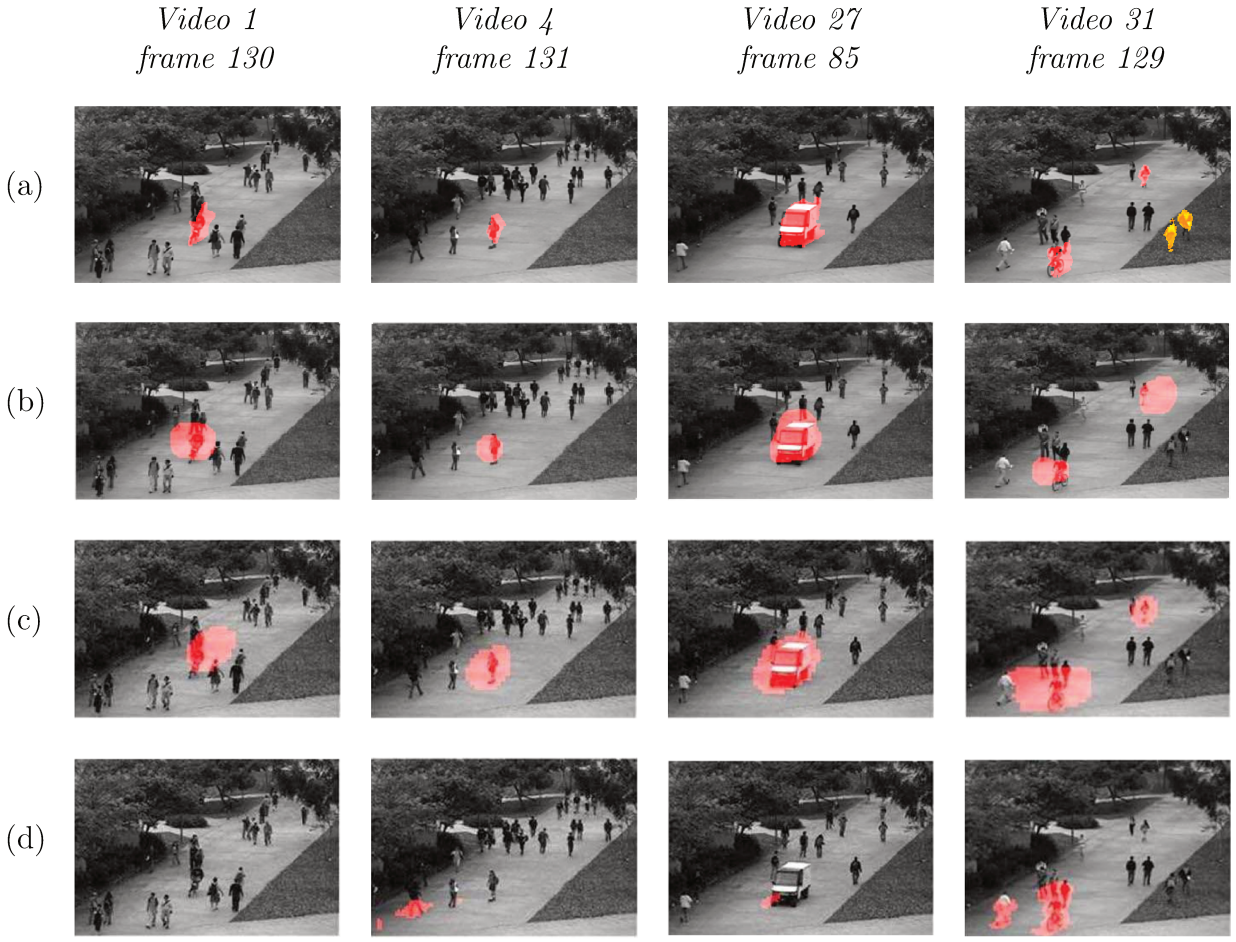


Figura 61 – *Resultados de localização (pixel-level) em (a) Sistema proposto, (b) Social Attribute (CHEN* et al.*, 2015), (c) MDT (MAHADEVAN* et al.*, 2010), e (d) MPPCA + SF (MAHADEVAN* et al.*, 2010).*

anormalidade na metodologia *frame-level* dentre os outros trabalhos correlatos, somente superada por (SABOKROU *et al.*, 2017). No trabalho de (SABOKROU *et al.*, 2017) são utilizados vários *auto-encoders* para obter automaticamente *features* que serão posteriormente utilizados em uma rede neural do tipo *deep-learning*. No entanto, *features* obtidos automaticamente não são facilmente interpretáveis, com o qual, não é possível identificar que tipo de comportamento foi classificado como anormal. Em contraste, a obtenção de diferentes *features*, permite a classificação da anormalidade segundo sua natureza, como é apresentada na última coluna da Figura 61(a) onde a mesma cena apresenta anormalidade por velocidade e por presença. Das Figuras 61 e 60 pode-se observar que o sistema proposto apresenta uma localização (*pixel-level*) mais precisa do que os outros trabalhos, incluindo (SABOKROU *et al.*, 2017).

Finalmente, todos os algoritmos de detecção de anormalidade desenvolvidos para este sistema tiveram a fácil atualização do modelo como uma das suas prioridades, possibilitando a inclusão de novo conhecimento de forma rápida e eficiente o qual não é facilmente possível com técnicas baseadas em redes neurais ou SVMs.

Na sequência serão apresentadas as conclusões do trabalho assim como propostas para melhorar o desempenho do sistema em trabalhos futuros.

CAPÍTULO 5

Conclusões e trabalhos futuros

Neste trabalho abordou-se o projeto e implementação de um sistema de vigilância assistido por computador que visa identificar quatro casos de comportamentos considerados como anormais em cenários com diferentes densidades de pessoas. O primeiro caso de anormalidade acontece quando um grupo de pessoas na cena está caminhando e subitamente todas elas começam a correr; o que possivelmente seja causado por um evento de interesse, como por exemplo uma explosão. O segundo caso de anormalidade é associada às mudanças nos padrões de deslocamento de pessoas na cena, bem seja na sua localização, na sua velocidade ou na sua direção de movimento. No terceiro caso, a anormalidade acontece quando um grupo de pessoas aparece na cena caminhando na mesma direção, e uma pessoa tenta atravessá-las deslocando-se em direção contrária. Finalmente, o quarto caso de anormalidade acontece quando ingressam na cena objetos que comumente não estão presentes nela, por exemplo carros no meio de uma travessia pedestre.

Levando em conta que os três primeiros casos de anormalidade dependem das informações de movimento dos objetos na cena, foi utilizado um algoritmo conhecido como fluxo óptico para obter ditas informações. O fluxo óptico permite calcular uma aproximação do campo vetorial gerado pelo movimento dos objetos a partir de dois *frames* consecutivos. Neste trabalho foram testados quatro algoritmos para obtenção do fluxo óptico, a saber: (BOUGUET, 2001), (FARNEBÄCK, 2003), (ZACH *et al.*, 2009) e (BROX *et al.*, 2004) e, já que a técnica proposta em (BROX *et al.*, 2004) apresentou a maior estabilidade nos diferentes cenários testados, ela foi escolhida para ser usada neste trabalho.

Uma característica que compartilham as atuais técnicas para obtenção de fluxo óptico é sua alta sensibilidade ao ruído nas imagens de entrada (FERMÜLLER *et al.*, 2001), assim, vídeos com baixa qualidade podem gerar informação de movimento em lugares onde este não acontece. Adicionalmente, existem cenários onde objetos como bandeiras ou árvores apresentam pequenos movimentos oscilatórios gerando informação de fluxo óptico sem relevância para o propósito deste trabalho. Visando reduzir o ruído e as informações irrelevantes de fluxo óptico, foi utilizado um algoritmo de subtração de

fundo, o qual modela a aparência dos objetos constantes na cena, conhecidos como *background*, e gera uma imagem binária, conhecida como *foreground*, contendo unicamente os objetos temporais ou seja, aqueles que não estão constantemente presentes na cena. De forma similar ao fluxo óptico, também foram testados vários algoritmos de segmentação de fundo presentes na literatura, a saber: (GODBEHERE *et al.*, 2012), (KAEWTRAKULPONG; BOWDEN, 2001), (ZIVKOVIC, 2004) e (ZIVKOVIC; HEIJDEN, 2006). O desempenho destes algoritmos nos casos testados foram similares, sendo escolhido o algoritmo de (ZIVKOVIC; HEIJDEN, 2006) por não requerer uma grande quantidade de *frames* iniciais para estabelecer o modelo do fundo. A máscara de *foreground* gerada pelo algoritmo de subtração de fundo é usada para filtrar os vetores de fluxo óptico irrelevantes para o sistema, reduzindo o tempo de processamento entre *frames*.

Alguns trabalhos na literatura não utilizam nenhum algoritmo de detecção de fundo, obtendo os objetos com movimento diretamente do fluxo óptico através de um *threshold* de magnitude, assim, vetores de fluxo óptico com magnitude inferior a dito *threshold* são ignorados pelo sistema. No entanto, experimentos realizados neste trabalho, mostraram que essa abordagem não é apropriada em cenas com perspectiva, devido a que valores altos do *threshold* ignorarão o movimento dos objetos mais afastados da câmera, enquanto valores baixos incluirão movimentos pequenos de objetos que pertencem ao *background*, assim, foi decidido manter o algoritmo de subtração de fundo, embora ele significasse maior tempo de processamento em cada *frame*.

Neste trabalho, diferentes *features* foram utilizados para gerar modelos utilizando uma abordagem não supervisionada, onde o modelo é obtido a partir de amostras unicamente com comportamento normal, em contraste com as abordagens supervisionadas, nas quais são necessárias amostras contendo situações tanto normais quanto anormais. A abordagem não supervisionada foi selecionada em virtude dos poucos exemplos de comportamento anormal encontrados em situações reais, onde, por definição, são raros ou poucos prováveis de acontecer.

Na literatura podem-se encontrar diversas técnicas, resumidas no Capítulo 2, que empregam *features* analíticas para gerar modelos usando abordagens não supervisionadas. Entre elas destacam-se as técnicas que utilizam o fluxo óptico para gerar *features* conhecidas como histograma de orientações do fluxo óptico (HOOF). Nestas técnicas o modelo de comportamento normal é geralmente obtido utilizando uma máquina de vetor suporte (SVM) de uma única classe. Foram identificadas duas limitantes com essas abordagens: a primeira é o fato do *feature* HOOF conter informações de velocidade, direção e aparência embutidas em um único vetor, o que impossibilita discriminar a variável que gerou a anormalidade; a segunda limitante é o fato do modelo gerado pela SVM não ser facilmente atualizável uma vez calculados os seus parâmetros, dita atualização precisaria armazenar todas as amostras anteriores, o que, em vídeos de longa duração, significa um

gasto enorme de recursos computacionais. Visando superar essas duas limitantes, neste trabalho foram obtidos diferentes *features* para cada tipo de variável que pudesse apresentar anormalidade, como a direção, a velocidade, a aparência, etc. Do mesmo modo, foram projetados algoritmos de modelagem com capacidade de ser atualizados constantemente.

Os diferentes algoritmos desenvolvidos neste trabalho foram agrupados em módulos. Um módulo é uma unidade de processamento de informação independente que se especializa na extração de *features* e na obtenção de um modelo para detectar e localizar um tipo específico de anormalidade. Dessa forma, o primeiro caso de anormalidade é detectado pelo primeiro módulo, o segundo caso pelo segundo módulo e assim até completar os quatro casos levados em conta neste trabalho. A arquitetura baseada em módulos permitiu trabalhar cada caso de forma independente, podendo atualizar os algoritmos em um módulo sem necessidade de modificar os restantes. Esta arquitetura também facilitou a adição do quarto módulo para a detecção de anormalidades por aparência, o qual não estava inicialmente contemplado neste trabalho, por não se basear nas informações fornecidas pelo fluxo óptico, sem precisar modificar os algoritmos previamente desenvolvidos. Além do anterior, esta arquitetura permite avaliar o desempenho de cada módulo de forma independente, o que facilitou a comparação dos resultados reportados nos trabalhos correlatos utilizando os mesmos critérios e os mesmos *datasets*.

Dos resultados apresentados no Capítulo 4 pode-se concluir que, para o primeiro caso de anormalidade, o algoritmo proposto neste trabalho detectou corretamente a mencionada anormalidade nos três vídeos de teste do *dataset* PETS2009 assim como em todos os vídeos do *dataset* UMN, superando na correta classificação de cada *frame* aos trabalhos correlatos com que foi comparado. O modelo obtido para este caso é baseado em uma distribuição Gaussiana cujos parâmetros são recalculados a cada novo *frame* produzindo um modelo altamente atualizável.

O segundo caso de anormalidade, anormalidade por direção, velocidade ou presença, é o caso mais trabalhado na literatura e do qual mais resultados estão disponíveis para comparação. Neste módulo para cada variável de movimento (presença, direção e velocidade) foi obtida uma *feature* diferente. Essa separação de variáveis permitiu elaborar modelos de diferente complexidade dependendo do tipo de variável, assim, por exemplo, o modelo obtido para detecção de anormalidade por presença é muito mais simples e rápido do que o algoritmo para detecção de anormalidade por velocidade. Os resultados obtidos neste módulo apresentaram o desempenho esperado quando foi testado no *dataset* UCSD. As curvas ROC apresentadas na Seção 4.4.2 demonstram que o algoritmo de classificação de comportamento anormal por presença obteve, nos piores casos, um valor EER (*Equal Error Rate*) de aproximadamente 10% no *dataset* UCSDped1 e de aproximadamente 18% no *dataset* UCSDped2. Enquanto o algoritmo de classificação por direção obteve um valor EER de aproximadamente 20%. No caso de anormalidade por

velocidade o classificador obteve um valor de EER de aproximadamente 18% para todos os vídeos do *dataset* UCSDped1 e um valor EER de aproximadamente 16% para todos os vídeos do *dataset* UCSDped2. Uma comparação com trabalhos correlatos da saída combinada dos três algoritmos deste módulo foi apresentada na Seção 4.8 onde se evidenciou que os algoritmos aqui propostos obtiveram um desempenho superior a quase todos os trabalhos correlatos, somente superado pelo trabalho de (SABOKROU *et al.*, 2017). No trabalho de (SABOKROU *et al.*, 2017) são utilizados *autoencoders* para obter as *features* de forma automática e uma arquitetura *deep-learning* para gerar o modelo de comportamento. Embora o desempenho de detecção neste *dataset* seja melhor no trabalho do (SABOKROU *et al.*, 2017) ele possui as duas limitantes já levantadas anteriormente, já que, após a fase de treinamento não é possível atualizar o modelo, e, dado que as *features* são obtidas automaticamente por um algoritmo, tampouco é possível interpretar e classificar o tipo de anormalidade. Essas duas limitantes foram superadas pelos algoritmos propostos neste trabalho, classificando cada tipo de anormalidade e tornando os modelos atualizáveis constantemente. Além de superar essas duas limitantes os algoritmos propostos mostraram um melhor desempenho na tarefa de localização da anormalidade quando comparadas com todos os outros trabalhos correlatos, incluído (SABOKROU *et al.*, 2017) como é evidenciado nas Figuras 61 e 60.

Os resultados obtidos na detecção da anormalidade que acontece quando uma pessoa tenta atravessar uma multidão caminhando em direção contrária da direção de movimento da multidão (caso 3) mostraram que o algoritmo proposto obteve bons resultados quando a densidade das pessoas na multidão é relativamente baixa. Em multidões com densidades de pessoas muito altas a pessoa indo em direção contrária não consegue se movimentar e portanto, não é gerada informação de fluxo óptico para essa pessoa, nesse caso o algoritmo não consegue detectar a anormalidade gerando altas taxas de falsos negativos.

No último caso, anormalidade por aparência, os resultados não foram os esperados já que o algoritmo conseguiu identificar objetos anormais de tamanho grande como carros, mas não conseguiu identificar objetos anormais menores como bicicletas ou cadeiras de rodas. Este fato provavelmente decorre do fato das *features* calculadas não representar corretamente todas as variáveis envolvidas nas características de aparência como a forma, o tamanho, a cor, etc. Este módulo é o único que não processa o fluxo óptico e obtém as *features* diretamente das imagens de entrada onde o espaço de características é bem maior do que no caso do fluxo óptico. Também evidenciou-se uma alta sensibilidade às mudanças dos parâmetros no algoritmo, especialmente no tamanho das regiões em que são divididas as imagens de entrada, produzindo resultados muito diferentes para pequenas variações deste parâmetro. Melhorias neste algoritmo serão propostas como trabalho futuro.

Em síntese, obter diferentes *features* para cada tipo de anormalidade permitiu ter

um maior controle nos algoritmos de obtenção dos modelos de comportamento normal, também permitiram classificar cada anormalidade por sua natureza, obtendo maior informação que os trabalhos correlatos atuais. Todos os algoritmos para obtenção de modelos de comportamento normal foram projetados para serem constantemente atualizados, permitindo adicionar novos padrões que não apareceram na etapa de treinamento do sistema, o que é uma limitante dos trabalhos atuais baseados em Redes Neurais ou SVM.

No desenvolvimento deste trabalho evidenciou-se uma ausência muito alta de *datasets* focados na detecção de comportamento anormal em multidões. Essa ausência de *datasets* foi uma das principais dificuldades encontradas para comparar os resultados obtidos utilizando vários cenários, condições e critérios. A maioria dos trabalhos correlatos utilizam unicamente o *dataset* UCSD para quantificar o desempenho de seus algoritmos devido a que ele é um dos poucos a possuir *ground-truth*. Utilizar um único *dataset* para testar os algoritmos traz a grande desvantagem de estar sujeito a desenvolver algoritmos que funcionam muito bem para esse *dataset* em particular, mas com desempenho inferior em outras locações ou condições.

Do trabalho desenvolvido podem-se enumerar as seguintes conclusões:

1. Foi realizada uma revisão dos trabalhos correlatos mais relevantes na área de detecção de comportamentos anormais em cenarios com multidões identificando suas vantagens e falências.

2. Foram propostos e desenvolvidos algoritmos focados na superação das falências identificadas na revisão da literatura nas situações descritas na Seção 1.1.1 testando-os em cenários com diferentes densidades de pessoas.

3. Os algoritmos propostos neste trabalho foram implementados e testados comparando seu desempenho com aqueles disponíveis nos trabalhos correlatos.

4. Foi implementada uma nova arquitetura por módulos que permite adicionar facilmente novos algoritmos para situações ou densidades de pessoas não contempladas neste trabalho.

5.1 Trabalhos futuros

Na sequência são enumerados alguns aspectos no sistema proposto que podem ser acrescentados ou modificados visando melhorar seu desempenho:

1. Propor novas abordagens para detecção de anormalidades por aparência, onde possam-se utilizar informações de variações temporais de textura, cor e/ou forma.
2. Criar *datasets*, virtuais ou reais, focados na identificação de anormalidades com densidade de pessoas altas fornecendo seu *ground-truth* para avaliação.
3. Realizar um estudo detalhado dos parâmetros ajustáveis de cada algoritmo apresentado neste trabalho, assim como a sensibilidade da resposta a suas variações.
4. Implementar os módulos propostos em uma arquitetura paralela, diminuindo o tempo de processamento de cada *frame*.

Referências

AGGARWALA, J. K.; RYOO, M. S. Human activity analysis: A review. *ACM Computing Surveys*, v. 43, n. 3, p. 16:1–16:40, 2011. Citado na página 28.

ALMEIDA, I. R. de; JUNG, C. R. Change detection in human crowds. *Conference on Graphics, Patterns and Images*, n. 26, p. 63 – 69, 2013. Citado 2 vezes nas páginas 35 e 75.

ANDRADE, E. L.; BLUNSDEN, S.; FISHER, R. B. Modelling crowd scenes for event detection. *18th International Conference on Pattern Recognition*, v. 1, p. 175 – 178, 2006. Citado na página 32.

BLEI, D. M.; NG, A. Y.; JORDAN, M. I. Latent dirichlet allocation. *Journal of Machine Learning Research*, v. 3, p. 993–1022, 2003. Citado na página 33.

BOUGUET, J.-Y. Pyramidal implementation of the lucas kanade feature tracker. *Intel Corporation, Microprocessor Research Labs*, 2001. Citado 2 vezes nas páginas 100 e 121.

BOUTTEFROY, P. L. M.; BEGHDADI, A.; BOUZERDOUM, A.; PHUNG, S. L. Markov random fields for abnormal behavior detection on highways. *Proceedings in 2nd European Workshop of the Visual Information Processing*, p. 149–154, 2010. Citado na página 30.

BOUWMANS, T.; BAF, F. E.; VACHON, B. Background modeling using mixture of gaussians for foreground detection - a survey. *Recent Patents on Computer Science*, v. 3, p. 219–237, 2008. Citado na página 117.

BROX, T.; BRUHN, A.; PAPENBERG, N.; WEICKERT, J. High accuracy optical flow estimation based on a theory for warping. *Proc. 8th European Conference on Computer Vision*, v. 4, p. 25–36, May 2004. Citado 2 vezes nas páginas 42 e 100.

CHAUDHRY, R.; RAVICHANDRAN, A.; HAGER, G.; VIDAL, R. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. Citado na página 38.

CHEN, D. Y.; HUANG, P. C. Motion-based unusual event detection in human crowds. *J. on Visual Communication and Image Representation*, v. 22, n. 2, p. 178–186, 2011. Citado na página 75.

CHEN, Y.; ZHANG, H.; CHENG, F.; YUAN, D.; YOU, Y. Abnormal behaviors detection using particle motion model. *Sixth International Conference on Graphic and Image Processing (ICGIP 2014)*, 2015. Citado 5 vezes nas páginas , 32, 96, 97 e 98.

COLLINS, R. T.; LIPTON, A. J.; KANADE, T.; FUJIYOSHI, H.; DUGGINS, D.; TSIN, Y.; TOLLIVER, D.; ENOMOTO, N.; HASEGAWA, O.; BURT, P.; WIXSON, L. *A System for Video Surveillance and Monitoring*. [S.l.], 2000. Citado na página 21.

DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, v. 1, p. 886–893, 2005. Citado 2 vezes nas páginas 38 e 62.

DEERWESTER, S.; DUMAIS, S. T.; FURNAS, G. W.; LANDAUER, T. K.; HARSHMAN, R. Indexing by latent semantic analysis. *Journal Of The American Society For Information Science*, v. 4, n. 6, p. 391–407, 1990. Citado na página 30.

DORETTO, G.; CHIUSO, A.; WU, Y. N.; SOATTO, S. Dynamic textures. *International Journal of Computer Vision*, v. 51, n. 2, p. 91–109, 2003. Citado 2 vezes nas páginas 33 e 38.

ELGAMMAL, A.; HARWOOD, D.; DAVIS, L. Non-parametric model for background subtraction. *6th European Conference on Computer Vision*, v. 2, p. 751–767, 2000. Citado na página 116.

FARNEBÄCK, G. Two-frame motion estimation based on polynomial expansion. *Proceedings of the 13th Scandinavian Conference on Image Analysis*, p. 363–370, 2003. Citado 2 vezes nas páginas 100 e 121.

FERMÜLLER, C.; SHULMAN, D.; ALOIMONOS, Y. The statistics of optical flow. *Computer Vision and Image Understanding*, v. 82, n. 1, p. 1–32, 2001. Citado na página 100.

FRADI Hajer; DUGELAY Jean-Luc. Sparse feature tracking for crowd change detection and event recognition. In: *ICPR 2014, 22nd International Conference on Pattern Recognition, 24-28 August 2014, Stockholm, Sweden*. Stockholm, SUÈDE: [s.n.], 2014. Citado 2 vezes nas páginas 35 e 75.

GODBEHERE, A. B.; MATSUKAWA, A.; GOLDBERG, K. Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation. *IEEE American Control Conference (ACC)*, p. 4305–4312, 2012. Citado na página 101.

HASAN, M.; CHOI, J.; NEUMANN, j.; ROY-CHOWDHURY, A. K.; DAVIS, L. Learning temporal regularity in video sequences. *Proceedings of IEEE Computer Vision and Pattern Recognition*, 2016. Citado 3 vezes nas páginas 38, 96 e 97.

HORN, B.; SCHUNCK, B. Determining optical flow. *Artificial Intelligence*, v. 17, p. 185–204, 1981. Citado na página 120.

IEAAIE. 2014. Disponível em: <http://bit.kuas.edu.tw/ieaaie14/computer_vision_competition.html>. Citado na página 69.

ITO, Y.; KITANI, K. M.; BAGNELL, J. A.; HEBERT, M. Detecting interesting events using unsupervised density ratio estimation. *Proceedings on Computer Vision Workshops and Demonstrations*, v. 3, p. 151–161, 2012. Citado na página 38.

JAIN, R.; NAGEL, H. H. On the analysis of accumulative difference pictures from image sequences of real world scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 1, n. 2, p. 206–214, 1979. Citado na página 113.

JOLLIFFE, I. Principal component analysis. *Springer Series in Statistics*, 2002. Citado na página 32.

KAEWTRAKULPONG, P.; BOWDEN, R. An improved adaptive background mixture model for real- time tracking with shadow detection. *Proc. 2nd European Workshop on Advanced Video Based Surveillance Systems*, 2001. Citado na página 101.

KALTSA, V.; BRIASSOULI, A.; KOMPATSIARIS, I.; HADJILEONTIADIS, L. J.; STRINTZIS, M. G. Swarm intelligence for detecting interesting events in crowded environments. *IEEE Transactions on Image Processing*, p. 2153 – 2166, 2015. Citado 3 vezes nas páginas 38, 96 e 97.

KIM, J.; GRAUMAN, K. Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates. *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. Citado 3 vezes nas páginas 38, 96 e 97.

KITAGAWA, G. Monte carlo filter and smoother for non-gaussian non- linear state space models. *Journal of Computational And Graphical Statistics*, v. 5, p. 1–25, 1996. Citado na página 30.

KNUTH, D. E. *Art of Computer Programming*. 3. ed. [S.l.]: Addison-Wesley Professional, 1997. v. 2. 232 p. Citado na página 73.

LI, W.; MAHADEVAN, V.; VASCONCELOS, N. Anomaly detection and localization in crowded scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 36, n. 1, p. 18 – 32, 2014. Citado na página 38.

LIN, J.; KEOGH, E.; LONARDI, S.; CHIU, B. A symbolic representation of time series, with implications for streaming algorithms. *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, p. 2–11, 2003. Citado na página 31.

LIPTON, A. J.; COLLINS, R. T.; KANADE, T. Introduction to the special section on video surveillance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 22, n. 8, p. 745 – 746, 2000. Citado na página 28.

LOWE, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, v. 60, n. 2, p. 91–110, 2004. Citado na página 31.

LUCAS, B. D.; KANADE, T. An iterative image registration technique with an application to stereo vision. *International Joint Conference on Artificial Intelligence*, p. 674 – 679, 1981. Citado na página 120.

MAHADEVAN, V.; LI, W.; BHALODIA, V.; VASCONCELOS., N. Anomaly detection in crowded scenes. *IEEE Conference on Computer Vision and Pattern Recognition*, p. 1975 – 1981, 2010. Citado 7 vezes nas páginas , 33, 38, 62, 96, 97 e 98.

MATTHIAS Clamer. 2012. Disponível em: <http://www.getimages.com/search/2/film?artist=matthiasclamer>. Citado na página 68.

MEHRAN, R.; OYAMA, A.; SHAH, M. Abnormal crowd behavior detection using social force model. *IEEE Conference on Computer Vision and Pattern Recognition*, v. 2, p. 935–942, 2009. Citado 5 vezes nas páginas 32, 33, 38, 96 e 97.

NASCIMENTO, J. C.; FIGUEIREDO, M. A. T.; MARQUES, J. S. Activity recognition using a mixture of vector fields. *IEEE Transavtions on Image Processing*, v. 22, n. 5, p. 1712–1725, 2013. Citado na página 38.

NIU, W.; LONG, J.; HAN, D.; WANG, Y.-F. Human activity detection and recognition for video surveillance. *IEEE International Conference on Multimedia and Expo*, 2004. Citado na página 30.

OLIVER, N.; ROSARIO, B.; PENTLAND, A. A bayesian computer vision system for modelling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 22, n. 8, p. 831–843, 2000. Citado na página 116.

PETS. 2009. Disponível em: <http://www.hitech-projects.com/euprojects/cantata/datasets-cantata/dataset.html>. Citado na página 65.

PITT, M. K.; SHEPHARD, N. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, v. 92, n. 446, p. 590–599, 1999. Citado na página 30.

POLUS, A.; SCHOFER, J. L.; USHPIZ, A. Pedestrian flow and level of service. *Journal of Transportation Engineering*, 1983. Citado 4 vezes nas páginas , 25, 26 e 35.

POPOOLA, O. P.; WANG, K. Video-based abnormal human behavior recognition—a review. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, v. 42, n. 6, p. 865–878, 2012. Citado na página 29.

REN, W.-Y.; LL, G.-H.; CHEN, J.; LIANG, H.-Z. Abnormal crowd beravior detection using beravior entropy model. *International Conference on Wavelet Analysis and Pattern Recognition*, p. 212 – 221, 2012. Citado na página 34.

RIBNICK, E.; ATEV, S.; MASOUD, O.; PAPANIKOLOPOULOS, N.; VOYLES, R. Real-time detection of camera tampering. *IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2006. Citado 2 vezes nas páginas 45 e 46.

Rogat Security Group. *CCTV Monitoring system*. 2011. Disponível em: <http://www.rogatsecuritygroup.com/en/images/stories/cctb.png>. Citado 2 vezes nas páginas e 20.

ROSHTKHARI, M. J.; LEVINE, M. D. Online dominant and anomalous behavior detection in videos. *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. Citado na página 38.

RYAN, D.; DENMAN, S.; FOOKES, C.; SRIDHARAN, S. Textures of optical flow for real-time anomaly detection in crowds. *8th IEEE International Conference on Advanced Video and Signal-Based Surveillance*, p. 230–235, 2011. Citado na página 38.

SABOKROU, M.; FAYYAZ, M.; FATHY, M.; KLETTE, R. Deep-cascade: Cascading 3d deep neural networks for fast anomaly detection and localization in crowded scenes. *IEEE Transactions on Image Processing*, v. 26, n. 4, p. 1992–2004, 2017. Citado 5 vezes nas páginas , 96, 97, 98 e 103.

STAUFFER, C.; GRIMSON, W. Adaptive background mixture models for real-time tracking. *Proceedings in IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, v. 2, 1999. Citado 4 vezes nas páginas 30, 31, 32 e 114.

SU, H.; YANG, H.; ZHENG, S.; FAN, Y.; WEI, S. Crowd event perception based on spatio-temporal viscous fluid field. *IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance*, p. 458 – 463, 2012. Citado na página 33.

TAO, M. W.; BAI, J.; KOHLI, P.; PARIS, S. Simpleflow: A non-iterative, sublinear optical flow algorithm. *Computer Graphics Forum*, v. 31, n. 2, 2012. Citado na página 121.

UCSD. 2013. Disponível em: <http://www.svcl.ucsd.edu/projects/anomaly/dataset.htm>. Citado na página 67.

UMN. 2006. Disponível em: <http://mha.cs.umn.edu/Movies/Crowd-Activity-All.avi>. Citado na página 66.

VARADARAJAN, J.; ODOBEZ, J. Topic models for scene analysis and abnormality detection. *IEEE 12th International Conference on Computer Vision Workshops*, p. 1338 – 1345, 2009. Citado na página 29.

WALHA, A.; WALI, A.; ALIMI, A. M. A system of abnormal behaviour detection in aerial surveillance. *IEEE International Conference on Information Assurance and Security*, 2013. Citado na página 31.

WANG, T.; SNOUSSI, H. Histograms of optical flow orientation for visual abnormal events detection. *IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, p. 13–18, 2012. Citado na página 33.

WANG, Y.; WANG, D.; CHEN, F. Abnormal behavior detection using trajectory analysis in camera sensor networks. *International Journal of Distributed Sensor Networks*, p. 9, 2014. Citado na página 31.

WREN, C.; AZARHAYEJANI, A.; DARRELL, T.; PENTLAND, A. Pfinder: real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 19, n. 7, p. 780 – 785, 1997. Citado na página 113.

XIANG, T.; GONG, S. Video behaviour profiling and abnormality detection without manual labelling. *Proceedings IEEE International Conference on Computer Vision*, v. 2, p. 1238 – 1245, 2005. Citado 2 vezes nas páginas 30 e 32.

XU, D.; SONG, R.; WU, X.; LI, N.; FENG, W.; QIAN, H. Video anomaly detection based on a hierarchical activity discovery within spatio-temporal contexts. *Neurocomputing*, v. 143, p. 144–152, 2014. Citado 3 vezes nas páginas 34, 96 e 97.

YANG, Y.; HE, X.; ZHOU, Y.; TENG, Q.; FU, Y. Detection and recognition of abnormal events in crowds based on visual technologies. *Journal of Information and Computational Science*, v. 11, n. 5, p. 1443– 1453, 2014. Citado na página 35.

ZACH, C.; POCK, T.; BISCHOF, H. A duality based approach for realtime tv-l1 optical flow. *Statistical and geometrical approaches to visual motion analysis*, p. 23–45, 2009. Citado na página 100.

ZHANG, D.; PENG, H.; HAIBIN, Y.; LU, Y. Crowd abnormal behavior detection based on machine learning. *Information Technology Journal*, v. 12, p. 1199–1205, 2013. Citado na página 34.

ZHONG, H.; SHI, J.; VISONTAI, M. Detecting unusual activity in video. *Proceedings of the 2004 IEEE computer society conference on Computer vision and pattern recognition*, p. 819–826, 2004. Citado na página 29.

ZIVKOVIC, Z. Improved adaptive gaussian mixture model for background subtraction. *ICPR*, 2004. Citado na página 101.

ZIVKOVIC, Z.; HEIJDEN, F. van der. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern recognition letters*, v. 27, n. 7, p. 773–780, 2006. Citado na página 101.

Anexos

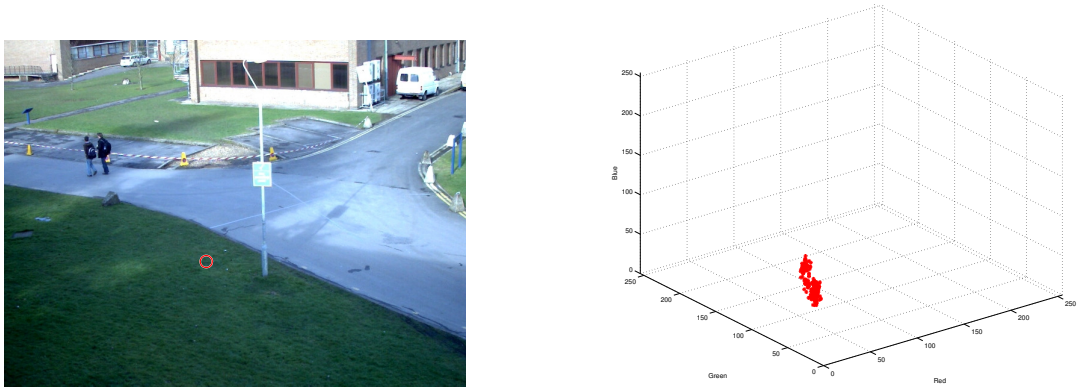
ANEXO A

Algoritmo de subtração de fundo

A subtração de fundo é uma etapa importante em muitos dos algoritmos de análise de vídeo. Em geral é o primeiro passo para segmentar objetos com movimento em cenas com câmeras fixas. A subtração de fundo é um procedimento por meio do qual são obtidos elementos que entram e saem da cena, é dizer, são segmentadas as regiões da cena que estão presentes de forma temporal, essas regiões são chamadas de *foreground* e as regiões fixas são chamadas de fundo ou *background*. Como a subtração de fundo é uma etapa muito usada nos algoritmos de rastreamento e detecção de objetos muitas técnicas têm sido desenvolvidas. Uma das primeiras abordagens apresentadas na literatura foi proposta por (JAIN; NAGEL, 1979), nesse artigo é feita uma diferença de *frames* de primeira ordem, onde o *frame* atual é subtraído do *frame* anterior. Assim é possível detectar elementos presentes no novo *frame* que não estavam presentes no *frame* anterior. A técnica desenvolvida nesse artigo apresenta bons resultados para entornos *indoor* mas é muito susceptível ao ruído e tem uma taxa de atualização muito alta, o que pode não ser desejado na maioria das aplicações. O principal problema do algoritmo de (JAIN; NAGEL, 1979) é a sensibilidade ao ruído e o fato de não levar em conta as variações de luminosidade.

A Figura 62 apresenta um exemplo das variações de um único *pixel* que pertence ao fundo da cena através de vários *frames*, é possível perceber que o mesmo *pixel* adota vários valores em uma pequena região no espaço de cores RGB.

Em (WREN *et al.*, 1997) é proposto um método que utiliza uma abordagem estatística para tentar modelar a situação descrita na Figura 62 (b). Cada *pixel* tem associada uma distribuição de probabilidades do tipo Gaussiano definida pelo vetor de médias ($\boldsymbol{\mu}$) e pela matriz de covariância ($\boldsymbol{\Sigma}$) do conjunto de valores da vizinhança. O modelo de cada *pixel* é atualizado periodicamente para adaptá-lo às mudanças na iluminação. Assim para cada novo valor de *pixel* determina-se se ele pertence ao modelo Gaussiano ou não. Caso o novo valor do *pixel* esteja dentro do modelo é dito que o *pixel* pertence ao fundo da cena (*background*), caso contrário o *pixel* não pertence ao modelo e é dito que pertence ao



(a) (b)

Figura 62 – Exemplo da variação de valores RGB em um único *pixel*. (a) Localização do *pixel* (337, 369) dentro da imagem original (centro do círculo vermelho). (b) Distribuição dos valores RGB adotados pelo pixel através do tempo.

foreground. A modelagem de cada *pixel* utilizando um modelo Gaussiano permite reduzir o número de falsos positivos e apresenta bons resultados em entornos exteriores como ruas e parques, porém utilizar uma única gaussiana resultou insuficiente em situações onde se apresenta movimento repetitivo como folhas de árvores ou ondas no mar.

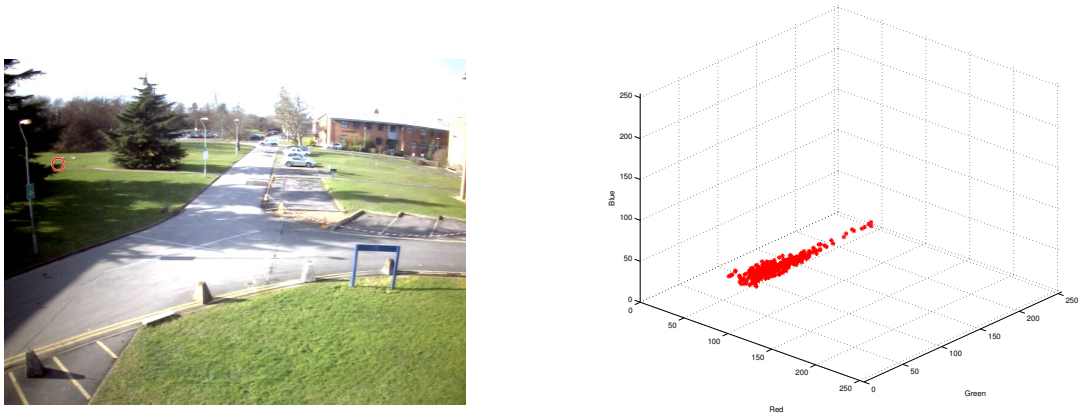
A Figura 63 apresenta um exemplo onde o método anterior apresenta falhas, em dita Figura são apresentados os valores RGB de um único *pixel* através do tempo, mas neste caso existe uma variação muito maior do que na apresentada na Figura 62 (b) devido à presença do galho da árvore que tem uma cor mais escura que a cor do gramado do fundo e seu movimento periódico geram uma distribuição como a apresentada na Figura 63 (b)

Visando superar as limitantes do método do Wren e incluir modelos para as distribuições como apresentadas na Figura 63 (b) foi proposto em (STAUFFER; GRIMSON, 1999) utilizar não uma, mas um número K de gaussianas para criar um modelo multinodal que se adapte melhor às condições de movimento repetitivo. A continuação será detalhado este método.

Seja o vetor X_i os componentes da cor do *pixel* (x_0, y_0) na imagem I_i em uma sequência de imagens I, o chamado historial do *pixel* é definido como

$$\{X_1, ..., X_t\} = \{I(x_0, y_0, i) : 1 \leq i \leq t\} \tag{A.1}$$

A história recente $\{X_1, ..., X_t\}$ de cada *pixel* é modelada como a mistura de K



(a) (b)

Figura 63 – Exemplo da variação de valores RGB em um único *pixel* onde apresentam se elementos com movimento repetitivo. (a) Localização do *pixel* (90, 176) dentro da imagem original (centro do círculo vermelho). Nesta região o vento movimenta o galho da arvore e gera mudanças repetitivas nos valores do *pixel* (b) Distribuição dos valores RGB adotados pelo pixel através do tempo.

distribuições Gaussianas. Assim a probabilidade de observar o valor atual X_t é dado por

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \tag{A.2}$$

onde K é o número de distribuições Gaussianas, $\omega_{i,t}$ é um valor estimado de peso ou relevância da i^{th} Gaussiana na mistura no tempo t, $\mu_{i,t}$ é o vetor de médias da i^{th} Gaussiana na mistura no tempo t, $\Sigma_{i,t}$ é a matriz de covariância da i^{th} Gaussiana na mistura no tempo t e η é a função de densidade de probabilidade dada por

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} \left|\Sigma\right|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t-\mu_t)^T \Sigma^{-1}(X_t-\mu_t)} \tag{A.3}$$

Por rações de custo computacional os autores do trabalho tomam os valores dos canais R, G e B, de cada *pixel* como variáveis independentes e com a mesma variância σ^2, portanto a matriz de covariância Σ é obtida como

$$\Sigma_{i,t} = \sigma_i^2 \boldsymbol{I} \tag{A.4}$$

onde $\boldsymbol{I}$ é a matriz identidade e σ_i^2 é a variância dos valores em qualquer dos canais R, G ou B da i^{th} Gaussiana na mistura.

Para o modelo se adaptar às mudanças na iluminação e incorporar novos elementos dentro do modelo é necessária uma atualização dos elementos estatísticos (σ, μ) e do valor

de peso (ω) de cada componente Gaussiano. Para atualizar o valor de $\omega_{i,t}$ utiliza-se a seguinte equação

$$\omega_{i,t} = (1 - \alpha)\omega_{i,t-1} + \alpha(M_{i,t}) \tag{A.5}$$

onde α é a taxa de aprendizado, quanto mais perto de um, mais rápida é a incorporação de nova informação no modelo. $M_{i,t}$ assume o valor de 1 se X_i pertence ao i modelo gaussiano e zero caso contrário. O valor da média e desvio padrão da i-ésima distribuição Gaussiana somente são atualizados se os valores X_i do pixel pertencem a essa distribuição, e são respectivamente obtidos a partir de

$$\mu_t = (1 - \rho)\mu_{t-1} + \rho X_t \tag{A.6}$$

$$\sigma_t^2 = (1 - \rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T(X_t - \mu_t) \tag{A.7}$$

onde

$$\rho = \alpha\eta(X_t|\mu_k, \sigma_k) \tag{A.8}$$

Cada pixel tem associadas de 3 a 5 distribuições gaussianas as quais são ordenadas de forma decrescente de acordo com o valor de ω/σ, onde as distribuições com maior peso e menor desvio padrão serão posicionadas nas primeiras colocações. As primeiras B distribuições são escolhidas para compor o modelo segundo a Equação A.9

$$B = argmin_b \left(\sum_{k=1}^{b} \omega_k > T \right) \tag{A.9}$$

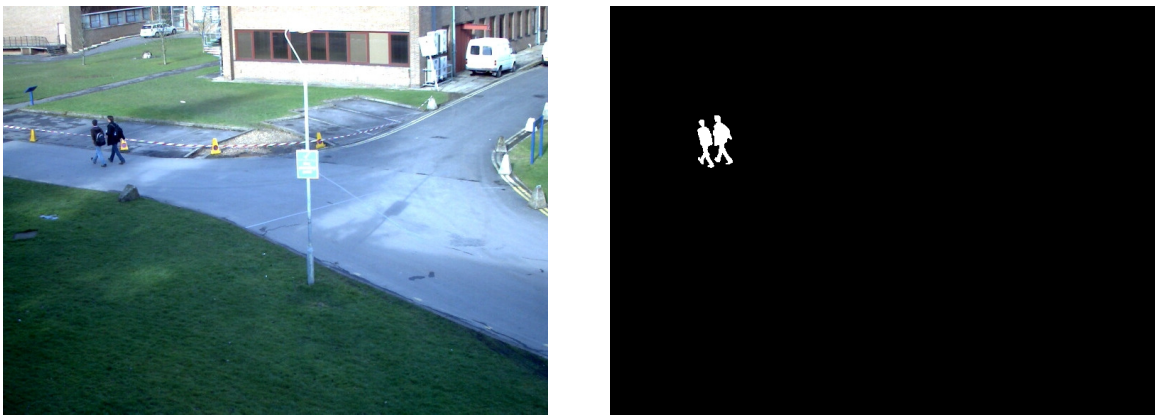
onde T é um parâmetro que permite ajustar a complexidade da mistura: um valor baixo de T levará em conta poucas distribuições e o modelo será uni-modal; para um valor alto de T são levadas em conta mais distribuições permitindo um modelo multi-modal, mas aumentando o custo computacional.

É possível encontrar na literatura muitas outras abordagens e algoritmos para subtração de fundo, por exemplo em (ELGAMMAL *et al.*, 2000) é apresentada uma técnica chamada de *Kernel Density Estimation* (KDE) que é uma generalização do algoritmo de mistura de Gaussianas. Neste método a intensidade de cada *pixel* é modelada como uma distribuição Gaussiana mas é utilizado um modelo não paramétrico e portanto diminui os erros obtidos pelo estabelecimento incorreto dos parâmetros requeridos pelo método da Mistura de Gaussianas. Em (OLIVER *et al.*, 2000) é apresentada uma técnica chamada de subtração do fundo por *Eigenbackground* onde é calculada a matriz de covariância de

um bloco de imagens de treinamento do fundo, depois é utilizado o PCA para reduzir a dimensionalidade do espaço e são obtidos os autovetores associados aos autovalores com maior valor. Esses autovetores formarão o modelo do fundo. Segundo os autores a técnica do *Eigenbackground* apresenta resultados parecidos aos obtidos utilizando a mistura de gaussianas porém com muita menos carga computacional.

A mistura de gaussianas é um algoritmo que apresenta várias vantagens frente a outros algoritmos: sua simplicidade, bons resultados e atualização constante fazem deste algoritmo o mais usado para a obtenção do *foreground* na maioria de trabalhos de detecção de comportamento anormal em vídeo e, portanto, será utilizado neste trabalho como primeira etapa de processamento do vídeo.

A Figura 64(b) apresenta a saída ideal dos algoritmos de subtração do fundo. Devido às simplificações feitas nos algoritmos, ruído nas imagens e parâmetros mal ajustados, a saída destes algoritmos geralmente não são tão perfeitas e apresentam tanto falsos positivos quanto falsos negativos. Em (BOUWMANS *et al.*, 2008) apresenta-se um *survey* das diferentes técnicas e melhoras ao algoritmo original de Mistura de Gaussianas com análise de desempenho e resultados obtidos pelos melhores algoritmos.



(a) (b)

Figura 64 – Saída ideal dos algoritmos de subtração de fundo. (a) Cena com duas pessoas caminhando. (b) Imagem binária (máscara) detectando os elementos temporais da cena.

A subtração de fundo é o primeiro passo em muitos algoritmos de processamento para segmentar os elementos de interesse e reduzir o custo gerado ao processar elementos que não são de interesse.

ANEXO B

Fluxo Óptico

O fluxo óptico é uma aproximação do campo vetorial de velocidades (conhecido como *Motion Field*) dos *pixels* em uma sequência de imagens, gerado pelo movimento relativo entre a câmera e os elementos da cena observada. Em outras palavras, o fluxo óptico determina quanto um *pixel* se movimento entre dois *frames* consecutivos. É possível obter fluxo óptico para imagens 2D e para volumes 3D, mas neste anexo só será levado em conta o caso 2D.

Foi dito que o fluxo óptico é uma aproximação ao campo de movimento real devido ao fato de precisar algumas simplificações para seu cálculo. Por exemplo o fluxo óptico assume que as mudanças no valor da intensidade de um *pixel* são devidas exclusivamente ao movimento dos objetos na cena. Também assume que os objetos da cena apresentam superfícies do tipo Lambertiana (superfície difusa), o que significa que a luz irradiada pelo objeto não depende do ângulo de visão do observador. Embora as simplificações necessárias, o fluxo óptico produz uma aproximação muito boa ao movimento real dos objetos na cena. A continuação será apresentada a modelagem matemática e os algoritmos para o cálculo do fluxo óptico.

Seja uma vizinhança de $n \times n$ *pixels* cujo centro está localizado no ponto (x, y). A intensidade I do *pixel* (x, y) é uma função da posição e do tempo $I(x, y, t)$. Suponha uma movimentação do objeto δx, δy em um tempo δt para $I(x + \delta x, y + \delta y, t + \delta t)$. Como $I(x, y, t)$ e $I(x + \delta x, y + \delta y, t + \delta t)$ são intensidades do mesmo ponto então tem-se

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t) \tag{B.1}$$

É possível realizar a expansão na Série de Taylor da parte direita de B.1 e obter,

$$I(x+\delta x, y+\delta y, t+\delta t) = I(x,y,t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t + T.O.S. \tag{B.2}$$

onde $T.O.S.$ são os Termos de Ordem Superior que podem ser desconsiderados sempre que o deslocamento δx e δy e o intervalo de tempo δt sejam pequenos. Assim de B.1 e B.2 obtém-se

$$\frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = 0 \tag{B.3}$$

$$\frac{\partial I}{\partial x}\frac{\delta x}{\delta t} + \frac{\partial I}{\partial y}\frac{\delta y}{\delta t} + \frac{\partial I}{\partial t}\frac{\delta t}{\delta t} = 0 \tag{B.4}$$

fazendo $\frac{\delta x}{\delta t} = v_x$, $\frac{\delta y}{\delta t} = v_y$ e como $\frac{\delta t}{\delta t} = 1$, de B.4 tem-se,

$$\frac{\partial I}{\partial x}v_x + \frac{\partial I}{\partial y}v_y + \frac{\partial I}{\partial t} = 0 \tag{B.5}$$

onde v_x e v_y são as componentes na direção $\bar{x}$ e $\bar{y}$ do fluxo óptico e $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$ e $\frac{\partial I}{\partial t}$ são as derivadas de intensidade no ponto (x, y, t). A Equação B.5 pode ser escrita de forma mais compacta como

$$\nabla I \cdot v^T + I_t = 0 \tag{B.6}$$

onde $\nabla I = \left(\frac{\partial I}{\partial x} + \frac{\partial I}{\partial y}\right)$ é o gradiente de intensidade espacial, $v = \left(v_x, v_y\right)$ é o vetor do fluxo óptico no *pixel* (x, y) no instante t e $I_t = \left(\frac{\partial I}{\partial t}\right)$ é o gradiente de intensidade temporal. A Equação B.6 é conhecida como a equação de restrição de movimento 2D.

O fato do cálculo do fluxo óptico ser baseado no gradiente de intensidade leva ao problema conhecido como problema de abertura. O problema de abertura aparece quando existe movimento em direção paralela à direção onde não existe mudança de intensidade. A Figura 65 apresenta um exemplo onde aparece o problema de abertura. Se uma barra movimenta-se na direção acima e à direita como indica o vetor $\vec{V}$ mas o movimento é observado somente através de uma pequena região, chamada abertura, o movimento da barra na direção $\vec{V_t}$ não é levada em conta no cálculo do fluxo óptico devido ao fato da intensidade da barra não mudar nessa direção. Por outro lado a componente na direção normal à barra, $\vec{V_n}$, possui toda a informação de movimento já que nessa direção o gradiente de intensidade é máximo.

Outra forma de explicar o problema de abertura é através do gráfico da Equação B.6 que é uma equação de duas incógnitas: uma linha recta no espaço (v_x, v_y). A Figura

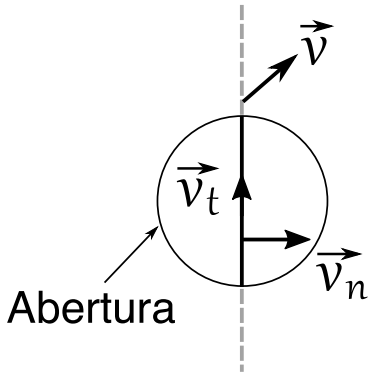


Figura 65 – Problema de abertura: somente é possível recuperar a informação de velocidade normal $\vec{v_n}$ onde existe mudança de intensidade.

66 apresenta a recta gerada pela Equação B.6, a velocidade correta está sobre a linha mas, pelo problema da abertura, só é possível obter a componente normal da velocidade real $\vec{v}_n$, o que se traduz como o vetor com menor magnitude da origem até a linha reta. A magnitude e a direção do vetor normal de velocidade $\vec{v}_n = v_n\hat{n}$ podem ser obtidas a partir das derivadas de intensidade I_x, I_y e I_t como:

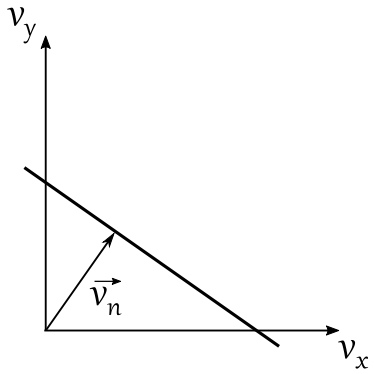


Figura 66 – A equação de restrição de movimento em 2D produz uma linha no espaço (v_x, v_y) onde a velocidade real é um ponto sobre essa linha, o vetor com menor magnitude desde a origem até a linha é o vetor normal v_n.

$$v_n = \frac{-I_t}{\|\nabla I\|_2} \tag{B.7}$$

e

$$\hat{n} = \frac{(I_x, I_y)}{\|\nabla I\|_2} \tag{B.8}$$

Existem várias técnicas para calcular o fluxo óptico, as duas mais famosas e usadas pela maioria dos trabalhos de detecção de comportamento anormal são os algoritmos desenvolvidos por Lucas-Kanade (LUCAS; KANADE, 1981) e Horn-Schunck (HORN; SCHUNCK, 1981). O primeiro deles realiza um cálculo de minimização do erro quadrático local da Equação B.6, o segundo realiza um cálculo de regularização global. A maior

diferença entre as duas técnicas é o que se conhece como fluxo óptico denso e fluxo óptico esparso. O algoritmo de Lucas-Kanade calcula o fluxo óptico de formal local (em uma vizinhança $n \times n$ de um *pixel* (x, y)) portanto é possível obter o fluxo óptico de um conjunto de *pixels* de interesse (fluxo óptico esparso). O algoritmo de Horn-Schunck realiza um cálculo global, utiliza todos os *pixels* da imagem (fluxo óptico denso) o que leva um maior custo computacional obtendo resultados mais precisos.

A partir dos trabalhos originais de Lucas-Kanade e Horn-Schunck muitas outras abordagens e melhoras têm surgido nos últimos anos. Alguns dos mais importantes são (BOUGUET, 2001) onde é apresentado um algoritmo baseado em Lucas-Kanade mas escalando cada *frame* consecutivamente criando uma implementação piramidal do algoritmo original. Essa implementação piramidal permite obter o fluxo óptico quando a velocidade de um *pixel* de um *frame* para o outro é muito alta. Em (FARNEBÄCK, 2003) é apresentada uma técnica para o cálculo do fluxo óptico denso usando uma expansão polinomial entre dois *frames* consecutivos. Em (TAO *et al.*, 2012) é apresentada uma técnica para o cálculo de fluxo óptico denso sem precisar da implementação iterativa do algoritmo original de Horn-Schunck o que diminui o tempo de cômputo e permite processar *frames* com resoluções maiores às alcançadas pelo algoritmo original. Todos estes algoritmos estão implementados na biblioteca aberta OpenCv.

A modelagem do fundo e a obtenção do *foreground* permitem a criação de uma máscara que pode ser usada para reduzir o ruído gerado pelo movimento de objetos pertencentes ao fundo da cena. Essa máscara é utilizada junto com o fluxo óptico como entrada a todos os módulos de detecção de comportamento anormal.