



UNIVERSIDADE ESTADUAL DE CAMPINAS  
Faculdade de Tecnologia

**Vinicius Destri Galafasse**

**Posicionamento *Indoor* utilizando comunicação  
*Screen-to-Camera (S2CP)***

Limeira  
2025

**Vinicius Destri Galafasse**

**Posicionamento *Indoor* utilizando comunicação *Screen-to-Camera*  
(S2CP)**

Dissertação apresentada à Faculdade de Tecnologia da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Mestre em Tecnologia, na área de Sistemas de Informação e Comunicação.

**Orientador: Prof. Dr. Leandro Ronchini Ximenes**

Este trabalho corresponde à versão final da Dissertação defendida por Vinicius Destri Galafasse e orientada pelo Prof. Dr. Leandro Ronchini Ximenes.

Limeira  
2025

Ficha catalográfica  
Universidade Estadual de Campinas (UNICAMP)  
Biblioteca da Faculdade de Tecnologia  
Felipe de Souza Bueno - CRB 8/8577

G13p Galafasse, Vinicius Destri, 1998-  
Posicionamento indoor utilizando comunicação Screen-to-Camera (S2CP)  
/ Vinicius Destri Galafasse. – Limeira, SP : [s.n.], 2025.

Orientador: Leandro Ronchini Ximenes.  
Dissertação (mestrado) – Universidade Estadual de Campinas  
(UNICAMP), Faculdade de Tecnologia.

1. Comunicação com luz visível. 2. Sistema de posicionamento em  
ambientes internos. 3. Processamento digital de sinais. I. Ximenes, Leandro  
R., 1985-. II. Universidade Estadual de Campinas (UNICAMP). Faculdade de  
Tecnologia. III. Título.

Informações complementares

**Título em outro idioma:** Indoor Screen-to-Camera Positioning (S2CP)

**Palavras-chave em inglês:**

Visible light communication

Indoor positioning systems

Digital signal processing

**Área de concentração:** Sistemas de Informação e Comunicação

**Titulação:** Mestre em Tecnologia

**Banca examinadora:**

Leandro Ronchini Ximenes [Orientador]

Rangel Arthur

Ricardo Barroso Leite

**Data de defesa:** 25-02-2025

**Programa de Pós-Graduação:** Tecnologia

**Objetivos de Desenvolvimento Sustentável (ODS)**

ODS: 9. Inovação e infraestrutura

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

- ORCID do autor: <https://orcid.org/0009-0001-8110-0673>

- Currículo Lattes do autor: <http://lattes.cnpq.br/0081893288524499>

## FOLHA DE APROVAÇÃO

Abaixo se apresentam os membros da comissão julgadora da sessão pública de defesa de dissertação para o Título de Mestre em Tecnologia na área de concentração Sistemas de Informação e Comunicação, a que se submeteu o aluno Vinicius Destri Galafasse, em 20 de fevereiro de 2025 na Faculdade de Tecnologia – FT/UNICAMP, em Limeira/SP.

**Prof. Dr. Leandro Ronchini Ximenes**

Presidente da Comissão Julgadora

**Prof. Dr. Rangel Arthur**

FT/UNICAMP

**Prof. Dr. Ricardo Barroso Leite**

IFSP

Ata da defesa, assinada pelos membros da Comissão Examinadora, encontra-se no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria de Pós-graduação da Faculdade de Tecnologia.

# Agradecimentos

Agradeço ao Professor Dr. Leandro Ronchini Ximenes por abraçar a ideia deste projeto e fornecer toda a orientação e suporte necessário como auxílio ao meu desenvolvimento.

Agradeço a Unicamp, Programa de Pós-Graduação em Tecnologia e o Laboratório de Processamento de Sinais pela oportunidade de crescer e desenvolver esta pesquisa com toda a infraestrutura e auxílio necessário.

Agradeço meus pais Denise Destri Galafasse e José Roberto Galafasse por todo o suporte, educação e carinho que sempre me deram, por acreditarem na minha capacidade e me ajudarem diretamente e indiretamente no meu desenvolvimento e neste projeto.

Agradeço a Jéssica Severiano Soares por todo o suporte e carinho que me ajudaram a passar por esta etapa com mais tranquilidade.

Agradeço a Deus pela minha saúde, força e motivação para sempre seguir em frente.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

# Resumo

Sistemas de posicionamento de equipamentos eletrônicos como *smartphones* e *drones* em ambientes internos desempenham um papel crucial em aplicações emergentes como realidade aumentada, realidade virtual, automação industrial e navegação em espaços fechados. Nesse contexto, as soluções baseadas em *Visible Light Positioning* (VLP) têm recentemente se destacado devido à profusão desses dispositivos no cotidiano.

Neste contexto, este trabalho propõe um sistema chamado de *Screen-to-Camera Positioning* (S2CP), baseado na exploração de telas digitais, como televisões e *displays* de uso geral, e câmeras digitais, como as encontradas em *smartphones* e *drones*, sendo único na literatura no uso desses dispositivos para as tarefas conjuntas de posicionamento e comunicações. Mais especificamente, o sistema desenvolvido utiliza um protocolo de comunicação que emprega códigos de barras e um protocolo baseado em decomposições tensoriais para transmitir dimensões de uma tela através de codificação espaço-temporal, permitindo ao receptor portador de uma câmera calcular sua posição relativa ao receptor de forma precisa para até aproximadamente 2 metros de distância, sem necessidade de triangulação ou trilateração adicional. O método proposto é compatível com uma única tela, envolve etapas de processamento de imagens, visão computacional e de decodificação de mensagens. Como resultados de validação, os experimentos realizados demonstraram precisão na ordem de poucos centímetros ( $\leq 10$  cm), destacando seu potencial em aplicações que requerem alta precisão e baixo custo operacional, especialmente em cenários cada vez mais repletos de telas digitais.

As perspectivas futuras a partir desta dissertação incluem cenários com múltiplas telas e sistemas com maior interação entre dispositivos inteligentes.

**Palavras-chave:** posicionamento por luz visível (VLP), comunicações por câmera óptica (OCC), tela para câmera (S2C)

# Abstract

Positioning systems for electronic devices such as smartphones and drones in indoor environments play a crucial role in emerging applications like augmented reality, virtual reality, industrial automation, and navigation in enclosed spaces. In this context, solutions based on Visible Light Positioning (VLP) have recently gained prominence due to the proliferation of these devices in everyday life.

This work proposes a system called Screen-to-Camera Positioning (S2CP), which leverages digital screens such as televisions and general-purpose displays, and digital cameras such as those from smartphones and drones. It is unique in the literature for using these devices for the combined tasks of positioning and communication. More specifically, the developed system uses a communication protocol that employs barcodes and a protocol based on tensor decompositions to transmit the dimensions of a screen through spatiotemporal encoding. This allows a receiver equipped with a camera to calculate its position relative to the transmitter accurately, for up to approximately 2 meters, without requiring additional triangulation or trilateration. The proposed method is compatible with a single screen and involves image processing, computer vision, and message decoding steps. Validation results demonstrated precision within a few centimeters ( $\leq 10$  cm), highlighting its potential in applications that require high accuracy and low operational cost, especially in environments increasingly filled with digital screens.

Future perspectives stemming from this dissertation include scenarios with multiple screens and systems with greater interaction among smart devices.

**Keywords:** visible light positioning (VLP), optical camera communications (OCC), screen-to-camera (S2C)

# Lista de Figuras

2.1	Exemplo de um sistema VLC . . . . .	22
2.2	Exemplo de um sistema OCC . . . . .	24
2.3	Exemplo de um sistema S2C . . . . .	26
2.4	Exemplo VLP . . . . .	28
2.5	Exemplo RGB . . . . .	29
2.6	Exemplos das principais operações morfológicas . . . . .	34
2.7	Exemplo do Detector de Bordas de Canny . . . . .	36
2.8	Exemplos de Homografia . . . . .	38
2.9	Exemplo de um Código de Barras 128 . . . . .	40
2.10	Exemplo PnP . . . . .	41
3.1	Ilustração do sistema proposto . . . . .	45
3.2	Fluxograma de visão geral . . . . .	47
3.3	Eixos do sistema da tela . . . . .	49
3.4	Padrão da Mensagem . . . . .	50
3.5	Fluxograma calibração . . . . .	52
3.6	Pontos no tabuleiro de xadrez . . . . .	53
3.7	Resultado pós calibração da câmera . . . . .	54
3.8	Pré processamento dos quadros . . . . .	56
3.9	Identificação da tela . . . . .	57
3.10	Contornos destacados . . . . .	58
3.11	<i>Bounding Boxes</i> de um quadro . . . . .	59
3.12	Quadro com código de barras . . . . .	61
3.13	Relações de distâncias e dimensões entre a câmera e a tela . . . . .	62
3.14	Exemplo P4P . . . . .	66
4.1	Demonstração do ambiente de gravação . . . . .	70
4.2	Comparação das coordenadas reais e estimadas . . . . .	76
4.3	Quadro da gravação 11 . . . . .	82
4.4	Posição plotada da câmera . . . . .	83
5.1	Organização dos quadros do protocolo tensorial . . . . .	86
5.2	Fluxograma adaptado para o protocolo tensorial . . . . .	87
5.3	Esboço do protocolo tensorial . . . . .	88
5.4	Quadro codificado após conversão ao protocolo tensorial . . . . .	89
5.5	Fluxograma da análise de variação de pixels . . . . .	90
5.6	Cena desenhada para simulações . . . . .	91
5.7	Cena simulada com múltiplas telas . . . . .	92
5.8	Tela detectada por variação temporal de pixels . . . . .	93
5.9	Homografia e recorte da tela detectada . . . . .	94

5.10	<i>Console</i> com informação da região de interesse detectada . . . . .	94
5.11	Fluxograma da detecção das matrizes codificadas . . . . .	95
5.12	Simulação do protocolo tensorial com perspectiva aplicada . . . . .	96
5.13	Recorte de um quadro do tensor após ajuste de perspectiva . . . . .	96
5.14	Dados retornados pelo decodificador do protocolo tensorial . . . . .	97
5.15	Demonstração do ambiente de gravação . . . . .	98
7.1	Códigos de caracteres BC-128 parte I . . . . .	116
7.2	Códigos de caracteres BC-128 parte II . . . . .	117
A.1	Quadro principal do editor de vídeo . . . . .	119
A.2	Editor Modo Deslizante . . . . .	120
A.3	Editor Modo Matriz Visual . . . . .	120
A.4	Editor Modo de Homografia Assistida . . . . .	121
A.5	Exemplo do Modo de Homografia Assistida . . . . .	121
A.6	Quadro transformado do vídeo . . . . .	122

# Lista de Tabelas

4.1	Resultados eixo X com BC-128 . . . . .	72
4.2	Resultados eixo Y com BC-128 . . . . .	73
4.3	Resultados eixo Z com BC-128 . . . . .	74
4.4	Resultados da Distância Euclidiana Calculada . . . . .	78
4.5	Resultados Distância - Detectada I . . . . .	79
4.6	Resultados Distância - Detectada II . . . . .	80
5.1	Resultados eixo X com protocolo tensorial . . . . .	99
5.2	Resultados eixo Y com protocolo tensorial . . . . .	100
5.3	Resultados eixo Z com protocolo tensorial . . . . .	100

# Lista de Abreviaturas e Siglas

AVI	<i>Audio Video Interleave</i>
BC-128	Código de Barras 128
BER	<i>Bit Error Rate</i>
CAP	<i>Carrierless Amplitude Phase Modulation</i>
CIR	Resposta ao Impulso do Canal - <i>Channel Impulse Response</i>
CMYK	<i>Cyan, Magenta, Yellow, Black</i>
CSK	<i>Color-Shift Keying</i>
FPS	Quadros por Segundo - <i>Frames per Second</i>
GPS	<i>Global Positioning System</i>
HSL/HSV	<i>Hue, Saturation, Lightness/Value</i>
IoT	<i>Internet of Things</i>
LiDAR	<i>Light Detection and Ranging</i>
LoS	Linha de Visada - <i>Line of Sight</i>
MIMO	<i>Multiple-Input and Multiple-Output</i>
MISO	<i>Multiple-Input and Single-Output</i>
OCC	<i>Optical Camera Communications</i>
OOK	<i>On-Off keying</i>
P3P	<i>Perspective-3-Point</i>
P4P	<i>Perspective-4-Point</i>
PD	Fotodetector - <i>Photodetector</i>
PWM	<i>Pulse Width Modulation</i>
QR	<i>Quick Response</i>
RGB	<i>Red, Green, Blue</i>
ROI	Região de Interesse - <i>Region of Interest</i>

RSS	Potência do sinal Recebido - <i>Received Signal Strength</i>
RTK	<i>Real Time Kinematic</i>
Rx	Receptor
S2CP	<i>Screen-to-Camera-Positioning</i>
SLAM	<i>Simultaneous Localization and Mapping</i>
Tx	Transmissor
UM-MIMO	<i>Ultra Massive Multiple Input Multiple Output</i>
VLP	<i>Visible Light Positioning</i>
VPPM	<i>Variable Pulse Position Modulation</i>

# Sumário

<b>1</b>	<b>Introdução</b>	<b>15</b>
1.1	Justificativas e Objetivos . . . . .	17
1.2	Organização do Documento . . . . .	20
<b>2</b>	<b>Fundamentos</b>	<b>21</b>
2.1	<i>Visible Light Communications (VLC)</i> . . . . .	22
2.2	<i>Optical Camera Communication</i> . . . . .	23
2.3	<i>Screen-to-Camera (S2C)</i> . . . . .	25
2.4	<i>Visible Light Positioning</i> . . . . .	26
2.5	Imagens digitais . . . . .	29
2.6	<i>Processamento de Imagens</i> . . . . .	30
2.7	Operações Morfológicas . . . . .	33
2.8	Detector de Bordas de Canny . . . . .	35
2.9	Homografia e perspectivas de imagens . . . . .	36
2.10	Código de Barras . . . . .	38
2.11	<i>Perspective-n-Point</i> . . . . .	40
2.12	Breve Revisão Literária . . . . .	41
<b>3</b>	<b>Desenvolvimento - S2CP com códigos de barras</b>	<b>44</b>
3.1	Transmissão do Vídeo Codificado . . . . .	48
3.2	Calibração da câmera . . . . .	50
3.3	Pré processamento dos quadros . . . . .	55
3.4	Identificação de regiões de interesse . . . . .	57
3.5	Identificação da tela . . . . .	59
3.6	Decodificação das mensagens para estimação da distância . . . . .	60
3.7	Obtenção da distância, pontos virtuais e reais . . . . .	61
3.8	Cálculo das coordenadas . . . . .	64
<b>4</b>	<b>Resultados - S2CP com códigos de barras</b>	<b>68</b>
4.1	Resultados usando sistema P4P . . . . .	71
4.2	Cálculo de distâncias entre câmera e tela . . . . .	78
4.3	Identificação e validação visual . . . . .	81
4.4	Precisão e medidas de erros . . . . .	84
<b>5</b>	<b>Adaptação do S2CP para Protocolo Tensorial</b>	<b>85</b>
5.1	Codificação da mensagem segundo esquema tensorial . . . . .	87
5.2	Identificação da Tela por Variação Temporal . . . . .	89
5.3	Identificação e decodificação do Tensor . . . . .	95
5.4	Testes reais com o protocolo tensorial . . . . .	97

<b>6</b>	<b>Conclusões</b>	<b>102</b>
<b>7</b>	<b>Perspectivas</b>	<b>104</b>
7.1	Posicionamento com Múltiplas Telas . . . . .	104
7.2	Flexibilização dos dados adicionais . . . . .	105
7.3	Otimização do esquema S2CP com protocolo tensorial . . . . .	105
7.4	Embarcar em Vídeo Real . . . . .	106
	<b>Referências bibliográficas</b>	<b>107</b>
	<b>Anexos</b>	<b>116</b>
<b>A</b>	<b>Editor de Perspectivas</b>	<b>118</b>
<b>B</b>	<b>Plano de Gestão de Dados</b>	<b>123</b>

# Capítulo 1

## Introdução

A área de Sistemas de Comunicação sem fio está em constante evolução e reconstrução de paradigmas. A cada lançamento de uma nova tecnologia, como o Wi-Fi, 3G, 4G e 5G é necessária a implementação de nova infraestrutura e a utilização de equipamentos compatíveis com as novas demandas. O crescimento dessa área acelerou muito nas últimas duas décadas se comparado aos avanços anteriores. (KHURPADE; RAO; SANGHAVI, 2018; VARSHNEY, 2012)

Atualmente as principais formas desses sistemas de comunicação se baseiam em radiofrequência. A faixa de RF comporta a faixa do espectro de ondas eletromagnéticas entre 3 kHz - 300 GHz e é classificada em diferentes categorias conforme a faixa dentro do espectro. Muitas tecnologias de comunicação e equipamentos eletrônicos utilizam a faixa conhecida como micro-ondas, variando entre 1 GHz - 300 GHz para aplicações de telecomunicações. (HOLKER, 1993)

Devido a esse intenso uso da banda de RF e micro-ondas, cada vez mais ocorrem problemas de interferência entre sinais transmitidos e recebidos para diferentes finalidades. Diversos métodos para mitigação dessas interferências e ganho na qualidade de transmissão já foram desenvolvidos, como diferentes tipos de modulação, equalização, seleção de banda de frequências e eliminação de ruídos. Porém, o constante crescimento da área caminha para um ponto em que o uso das microondas ficará saturado e todos esses métodos desenvolvidos não serão mais suficientes para a incorporação efetiva de novas tecnologias. Sendo assim, iniciou-se uma busca por alternativas para os futuros sistemas de comunicação principalmente na última década. (HAAS et al., 2019)

Tecnologias para Sistemas de Comunicação por Luz Visível - VLC vêm sendo estudadas e exploradas com crescente força desde a década passada devido ao seu potencial de aplicações para as redes 5G que estão sendo instaladas pelo mundo (KHURPADE; RAO; SANGHAVI, 2018), e principalmente para a futura rede 6G (TELI et al., 2022), que apresentará uma integração de comunicação e sensoriamento, automação e operação ainda maior dos dispositivos eletroeletrônicos através da chamada Internet das Coisas - *Internet of Things* (IoT). (JIANG et al., 2021)

O apelo da utilização de ondas de luz visível se deve a diversos fatores, dos quais destacam-se: possuir grande largura de banda no espectro eletromagnético, entre 400 THz e 800 THz (380 nm a 700 nm de comprimento de onda), não sofre interferência de nenhum dos outros sinais de comunicação já existentes em RF, pode ser incorporado em conjunto com sistemas de iluminação LED já utilizados dentro dos ambientes modernos, pode-se aproveitar das câmeras já presentes em inúmeros equipamentos eletrônicos, como os smartphones e drones, e até mesmo utilizar telas de computadores, televisão e outros tipos de *displays* para transmitir informações. (ZHANG, Z. et al., 2019)

Outro fator a favor do VLC é não provocar interferência eletromagnética em equipamentos eletrônicos, como os presentes em ambientes hospitalares e aviões. Além disso, traz benefícios em relação à segurança, pois os sinais luminosos não atravessam superfícies opacas, enquanto que ondas de RF podem ser mais facilmente espionadas por um equipamento próximo pois penetram diferentes superfícies. (RAJ et al., 2023)

Dentro das aplicações estudadas para o uso da VLC, há os sistemas de posicionamento e rastreamento *indoor* para drones e outros robôs autônomos, conhecidos como Posicionamento por Luz Visível - *Visible Light Positioning* (VLP). Diferentemente dos sistemas de *Global Positioning System* (GPS) utilizados para localização *outdoor*, dentro de espaços cobertos não é possível contar com o sinal e a precisão fornecida por satélites, pois a precisão dessa tecnologia que gira em média de 3 a 6 metros sofre perdas ao atravessar paredes e outras barreiras em ambientes internos, não suprindo as escalas de poucos centímetros de precisão e exatidão necessárias para a localização *indoor* (VIEIRA et al., 2024).

Além da tecnologia VLC com fotodetectores, há sistemas que empregam câmeras como receptores, chamados de *Optical Camera Communications* (OCC). (SAEED et al., 2019) Câmeras são capazes de capturar quadros que contém mais informações além de sinais luminosos, o que é benéfico para aplicações de posicionamento onde pode-se aproveitar

dessas dimensões capturadas para obter uma melhor precisão de separação espacial quando comparada ao uso de LEDs discretos. (ZHANG, P. et al., 2023) Alguns dispositivos possuem capacidades de processamento de imagem integradas, o que permite a aplicação de algoritmos avançados de visão computacional para extrair informações úteis do ambiente, e mesmo que não possuam, podem ser facilmente integradas com computadores externos para processamento. Além disso, as câmeras podem servir a múltiplos propósitos além do posicionamento, como captura de vídeo, reconhecimento de objetos e detecção de movimento.

Sendo assim, há a motivação de explorar as tecnologias de VLC e OCC, as quais apresentam maior precisão pois a luz visível sofre menos propagação de multipercurso e possui comprimentos de onda menores quando comparada ao RF. Estas implementações podem ser feitas de maneira isolada ou trabalhando de forma híbrida, como sistemas baseados em *bluetooth*. (SESYUK; IOANNOU; RASPOPOULOS, 2022)

*Screen-to-Camera (S2C)*, uma tecnologia PHY-VI OCC do padrão IEEE 802.15.7 (IEEE STANDARD ASSOCIATION, 2011), e mais recentemente no rascunho P802.15.7a/D8 (IEEE..., 2024), é uma tecnologia OCC relativamente pouco explorada que pode ser útil tanto para posicionamento quanto para comunicações de alta velocidade (ASHOK et al., 2015). No lado transmissor, telas digitais consistem em um número muito grande de pixels. Esse fator permite a caracterização de um sistema de comunicação *Ultra Massive Multiple Input Multiple Output (UM-MIMO)* (IEEE..., 2024; NIE; JORNET; AKYILDIZ, 2019), já que cada pixel na tela atua como um elemento transmissor. Quaisquer ganhos de diversidade espacial (transmissão) podem melhorar a robustez contra erros de símbolo causados por distorção do canal e problemas específicos, como pixels mortos. Além disso, múltiplos elementos transmissores também podem fornecer ganhos de multiplexação, aumentando a taxa de dados em comparação com os sistemas OCC típicos.

## 1.1 Justificativas e Objetivos

Um sistema de posicionamento *indoor* preciso é fundamental para uma ampla gama de aplicações modernas, como navegação e mapeamento em veículos autônomos, tanto terrestres quanto aéreos, e sistemas de realidade aumentada, pensando nas tendências da *IoT* e do 6G. Em veículos autônomos, como *drones*, a capacidade de determinar a localização

exata é essencial para o planejamento de rotas, prevenção de colisões e interação segura com o ambiente (THRUN; BURGARD; FOX, 2005; ELHOUSNI; HUANG, 2020). Da mesma forma, em aplicações de realidade aumentada, o posicionamento é crítico para alinhar objetos virtuais ao ambiente físico com alta precisão, garantindo uma experiência imersiva e funcional (AZUMA, 1997; ARTH et al., 2011). Técnicas como *Simultaneous Localization and Mapping* (SLAM), uso de sensores *Light Detection and Ranging* (LiDAR), câmeras e sinais GNSS têm desempenhado um papel central na evolução dessas tecnologias, abordando os desafios de precisão e robustez em ambientes dinâmicos e complexos. (DURRANT-WHYTE; BAILEY, 2006; GALLEGO; DELBRÜCK et al., 2020).

O escopo desta pesquisa é na investigação pioneira do potencial da tecnologia S2C para aplicações de posicionamento em ambientes *indoor* através de câmeras digitais em conjunto com a utilização de telas digitais como transmissores de dados. Esse tipo de sistema é batizado aqui como *Screen-to-Camera-Positioning* (S2CP). No entanto, até onde é do conhecimento dos autores, nenhum outro trabalho combinou S2C e VLP até agora, sendo (FENG et al., 2023) o mais próximo desse objetivo (mas com uma lâmpada LED gravada em vez de uma tela LED, e com o auxílio de um giroscópio do *smartphone*, ou seja, o sistema consta com outros sensores além da câmera para realizar o posicionamento).

Atualmente é ampla a presença de telas de computador, televisão e telões em ambientes residenciais, industriais, hospitalares, escolares, em comércios, supermercados, consultórios, entre outros, o que traz uma oportunidade de empregar esses equipamentos para usos de comunicação e posicionamento. (ZHANG, Z. et al., 2019)

Além da vantagem de mudar facilmente a posição do transmissor, substituir a lâmpada fixa por uma tela digital, como uma televisão ou *digital display*, permite a transmissão de informações adicionais em taxas de dados ainda mais altas, como dados de identificação, sequências de vídeo para visualização humana e sinais de controle para veículos autônomos ou robôs (BOIXADER et al., 2024).

Diferentemente de sistemas de posicionamento em VLC que utilizam lâmpadas e fotodetectores pontuais para captura de sinais transmitidos, ou de sistemas convencionais de posicionamento OCC que utilizam apenas fontes de luz discretas como transmissores e *beacons*, um sistema S2CP permite a utilização de toda a área da tela para transmissão de dados de forma paralela, seja para ganho na taxa de transmissão, ou para maior robustez com protocolos de redundância e correção de erros.

Telas digitais cuja função principal é transmitir imagens para a visão humana são usualmente fixadas horizontalmente nas paredes, portanto a referência de um sistema S2C é diferente de um VLC/OCC que identifica lâmpadas no teto. Além disso, com a geometria regular das telas, pode-se agregar técnicas de visão computacional para determinação de perspectivas e reconhecimento dos equipamentos para auxiliar no processo de posicionamento.

Por tratar-se de um sistema inédito, esta pesquisa explorou também a incorporação de um protocolo de comunicações que se adequasse à forma de transmissão e captura dos vídeos, usando um novo código espaço-temporal para aproveitar-se da sequência de quadros de vídeo.

Em resumo, este trabalho traz as seguintes contribuições e benefícios:

1. Introduz o primeiro método de posicionamento baseado na tecnologia S2C (S2CP), onde uma tela digital pode cumprir três objetivos principais: seu objetivo primário (transmissão visual) e duas novas funções: como referência de posicionamento para navegação em ambientes internos e como transmissor de OCC para comunicação de dados binários.
2. O S2CP utiliza um protocolo programável para posicionamento em tempo real, adaptável a qualquer tamanho, resolução ou posição da tela, oferecendo maior flexibilidade em comparação com dispositivos fixos como lâmpadas de LED ou marcadores impressos em papel, como códigos de barras/ *Quick Response* (QR).
3. Telas digitais permitem transmitir dados de posicionamento com impacto visual mínimo, evitando marcadores fixos como códigos QR montados na parede, o que é ideal para uso interno residencial. Diferentemente dos métodos convencionais de posicionamento baseados em QR, que frequentemente utilizam métodos de triangulação ou trilateração, o S2CP utiliza apenas alguns quadros para resolver suas equações de posicionamento. (PAN et al., 2020; KIM et al., 2021; KANG; CHOI; JUNG, J.-W., 2022)
4. Ao longo do desenvolvimento deste trabalho, criou-se um *Software* próprio para edições de perspectivas de vídeos, uma ferramenta que demonstrou-se muito útil para realizar pequenas correções e ajustes de perspectivas em gravações reais, e também para manipular simulações de gravações. Tal ferramenta encontra-se no Apêndice A.

## 1.2 Organização do Documento

O Capítulo 2 explica mais detalhadamente os conceitos principais utilizados neste trabalho, e apresenta também um breve levantamento bibliográfico com a contextualização do atual estado da arte para sistemas de posicionamento com tecnologias VLC/OCC/S2C.

O Capítulo 3 explica o método e desenvolvimento realizado para o sistema de posicionamento S2CP proposto. O Capítulo 4 agrupa os resultados e discussões da estimação do posicionamento, onde também são comparados tais resultados com um trabalho que aborda de maneira similar a identificação de uma lâmpada retangular no teto para posicionamento.

O Capítulo 5 traz uma adaptação ao sistema onde substituímos a utilização dos códigos de barras do Capítulo 3 por um protocolo inédito em desenvolvimento pela equipe do Laboratório de Processamento de Sinais da FT-Unicamp, que garante maior flexibilidade na estruturação da transmissão de dados.

O Capítulo 6 apresenta uma breve conclusão do projeto desenvolvido e dos resultados obtidos, validando o funcionamento e inovação obtidos. Por fim, o Capítulo 7 traz algumas linhas de investigação para futuros trabalhos que possam contribuir e seguir com esta pesquisa.

# Capítulo 2

## Fundamentos

Segundo as tecnologias do padrão IEEE, as taxas de transmissão atingidas pela comunicação óptica sem fio (*Short-Range Wireless Optical Communication using Visible Light*) são, em sua maioria, inferiores às de RF. É possível encontrar detalhadamente todas as especificações das categorias de transmissão definidas pela normativa do IEEE-802.15.7 (IEEE STANDARD ASSOCIATION, 2011). Essa divisão ocorre conforme a camada física do sistema:

- PHY-I: voltada para aplicações outdoor com taxas de dezenas a centenas de Kbps, utiliza modulações *On-Off keying* (OOK), *Variable Pulse Position Modulation* (VPPM), ou *Pulse Width Modulation* (PWM).
- PHY-II: uso indoor com taxa de transmissão de dezenas de Mbps e modulação OOK, VPPM, ou *Carrierless Amplitude Phase Modulation* (CAP).
- PHY-III: destinada a sistemas *Multiple-Input and Multiple-Output* (MIMO) com modulação *Color-Shift Keying* (CSK) e taxas de dezenas de Mbps.
- PHY-IV: fontes de luz discretas com até 22 Kbps de taxa de transmissão e múltiplas formas de modulações, como PWM e OOK.
- PHY-V: aplicada em fontes de luz para superfícies difusas com taxa de até 5,71 Kbps e diferentes modulações, como VPPM e PWM.
- PHY-VI: definida para o uso em displays com transmissão em poucos Kbps e variadas modulações, incluindo OOK e CSK. Este trabalho se enquadra nesta categoria.

A normativa estabelece esses padrões e os atualiza periodicamente, porém por tratar-se de uma área emergente e em constante inovação, muitos desenvolvimentos podem não se

enquadrar adequadamente sob essas definições, ou ainda provocar a necessidade de novas revisões. Desde sua primeira versão em 2011 à revisão mais recente em Julho de 2024 (publicada em Dezembro de 2024) (IEEE..., 2024), diversos avanços foram realizados em sistemas de comunicação óptica sem fio, sendo que já há estudos que atingiram taxas de 10 Gbps em laboratório. (XIE et al., 2020)

## 2.1 Visible Light Communications (VLC)

A Comunicação por Luz Visível (*Visible Light Communication - VLC*) é uma tecnologia de transmissão de dados que utiliza a luz visível emitida por LEDs para enviar informações. Ela funciona modulando a intensidade da luz em frequências imperceptíveis ao olho humano, permitindo que dispositivos equipados com sensores fotossensíveis recebam e decodifiquem os dados (PATHAK et al., 2015). Essa tecnologia oferece alta taxa de transmissão, segurança devido à limitação física da luz, e pode ser usada em ambientes onde as frequências de rádio são restritas, como hospitais e aeronaves. Na Figura 2.1 é demonstrado um enlace de VLC, onde ocorre a comunicação unidirecional entre um transmissor de luz visível e um Fotodetector - *Photodetector* (PD) que recebe os sinais luminosos sob determinada distância e angulação.

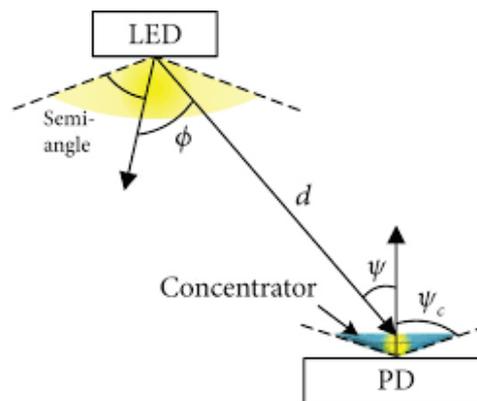


Figura 2.1: Exemplo simplificado de um sistema VLC, sendo a lâmpada o transmissor e o fotodetector (PD) o receptor, posicionados a uma distância  $d$ . O concentrador é a lente do PD,  $\phi$  e  $\Psi$  são os ângulos de emissão e incidência, respectivamente, enquanto que  $\Psi_c$  é o campo de visão do PD (MAPUNDA et al., 2020)

A pesquisa e desenvolvimento com VLC enfrentam desafios específicos. O bloqueio de sinais luminosos fora de uma Linha de Visada - *Line of Sight* (LoS) é uma vantagem em termos de segurança, pois impede o vazamento e a espionagem dos sinais por dispositivos

externos ao ambiente. No entanto, essa mesma característica exige maior cuidado no posicionamento dos equipamentos transmissores e receptores, já que variações na distância e no ângulo entre eles podem causar degradações significativas no canal de comunicação. Alguns trabalhos apresentam soluções com sistemas que não dependem de LoS, como em (MARTÍNEZ-CIRO et al., 2022). Além disso, outros fatores como a interferência entre diferentes fontes de iluminação, ruídos da luz ambiente e a capacidade de reflexão de diferentes cores em diferentes materiais ou texturas do ambiente também afetam a qualidade do enlace. (O'BRIEN et al., 2008)

O efeito de *flickering*, ou cintilamento, também é um obstáculo que deve sempre ser mitigado nas tecnologias de VLC. Tal efeito ocorre quando as variações da iluminação provocadas pela modulação empregada na transmissão se tornam perceptíveis ao nosso olho. Iluminação instável pode provocar sérios problemas cognitivos ao ser humano, portanto a VLC deve ser desenvolvida como uma função secundária dos equipamentos primariamente usados para iluminação. (PATHAK et al., 2015)

## 2.2 *Optical Camera Communication*

Há uma linha de tecnologias que utilizam comunicação por luz visível conhecida por *Optical Camera Communications* (OCC) (BAI et al., 2019; LI et al., 2018). Nesta, é empregado o uso de câmeras como receptores ao invés dos fotodetectores. Essa mudança possui vantagens e desvantagens. Tratando-se de seu uso em posicionamento *indoor*, uma das principais vantagens é a presença de câmeras já instaladas nos veículos autônomos (Ex: *drones*) e outros equipamentos eletrônicos móveis, como *smartphones* e *webcams*, não provocando aumento de custo nem de alterações no *hardware* dos produtos. Na Figura 2.2 é demonstrado um enlace de OCC, onde ocorre a comunicação unidirecional entre uma lâmpada de LED transmissora comumente encontrada em ambientes *indoor* urbanos, e o sensor/câmera receptora de um *laptop*.

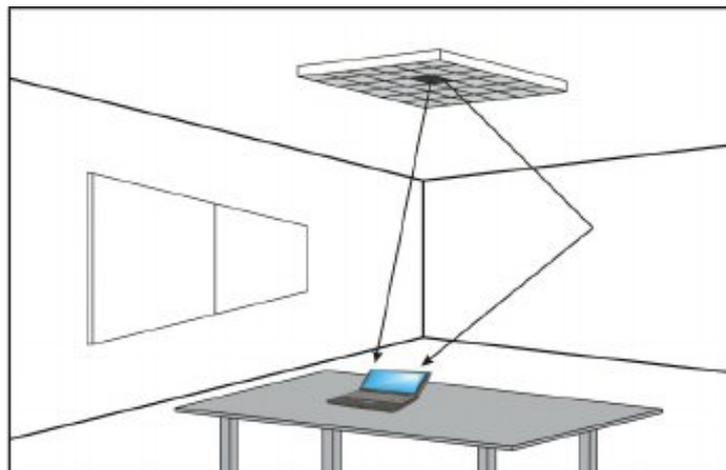


Figura 2.2: Exemplo simplificado de um sistema OCC, sendo a lâmpada o transmissor e a câmera do *laptop* o receptor. (BOUCHET et al., 2011)

Outra vantagem das câmeras é permitir a captura de imagens e vídeos para trabalhar com os sinais recebidos também no domínio espacial, permitindo inclusive o emprego de algoritmos de processamento de imagens para expandir as possibilidades de transmissão, diferentemente dos PDs que funcionam como um ponto de recepção e operam apenas com as potências e frequências recebidas.

A grande quantidade de fotodiodos presentes em sensores de imagem também é um diferencial que permite um maior grau de precisão e separação espacial dos elementos transmissores se comparado aos PDs avulsos. Quanto maior a resolução da câmera, maior a capacidade de segmentar o quadro e identificar os elementos de interesse, neste caso, os pixels referentes ao Transmissor (Tx).

Uma das principais desvantagens, porém, é a limitação das taxas de transmissão devido à velocidade de captura das câmeras. Em equipamentos de fácil acesso comercial, essas taxas de captura se dão em torno de 30 a 60 Quadros por Segundo - *Frames per Second* (FPS). É inviável utilizar esses valores em aplicações com altas demanda de taxa de transmissão, pois além de serem velocidades muito baixas para transmissão de dados, trabalhar com modulações em baixas frequências da portadora provoca efeitos de *flickering* perceptíveis ao olho humano. (SAEED et al., 2019).

Além da baixa taxa de transmissão oriunda da taxa de quadros por segundo de câmeras populares, há redução da taxa também devido a perdas de sincronização dos quadros, necessitando de novas estratégias para corrigir a sincronia. Dependendo do dispositivo que embarca a câmera, como *smartphones*, pode ser que esta já possua embutido também algum

código de ajuste adaptável de quadros, distorção, cores, saturação, entre outros, o que insere ainda mais efeitos de degradação que devem ser levados em consideração.

A solução adotada para viabilizar a OCC com equipamentos *low profile* é a exploração do efeito de *rolling shutter* das câmeras. Cada quadro capturado é composto pelo agrupamento e processamento da varredura vertical das linhas de pixels da câmera. É possível aproveitar esse funcionamento para transmitir os dados em taxas compatíveis com essa varredura, permitindo assim trabalhar com portadoras de frequências muito superiores a nossa percepção visual, atingindo dezenas de Kbps de velocidade de transmissão. (HAMAGAMI et al., 2021; LIN, B. et al., 2017; TRAN et al., 2018)

Contudo, ao trabalhar com câmeras para tarefas de posicionamento baseadas em visão computacional, as informações processadas não são completamente fieis ao espaço físico real observado. Uma câmera como sensor de imagem insere distorções em sua captura, provindas do material e geometria de sua(s) lente(s) (QUINTANA et al., 2020). Toda câmera é dimensionada com determinado comprimento focal e centro óptico. Em equipamentos digitais, há também o tamanho e quantidade de pixels para influenciar no resultado final. Tais variáveis são chamadas de parâmetros intrínsecos de uma câmera. Já os fatores externos à câmera que definem sua posição e orientação no espaço são conhecidos como parâmetros extrínsecos. (LU, 2018; PIXEL, 2024)

Os parâmetros intrínsecos permitem mapear as posições de pixels em uma imagem dentro do sistema de coordenadas da câmera. Os parâmetros extrínsecos permitem definir a localização e orientação da câmera no mundo real. Ambos são utilizados para formar um modelo onde é relacionado a posição real 3D de elementos à suas posições virtuais 2D no sistema de coordenadas da câmera, após devidas calibrações e correções da imagem obtida.

### **2.3 *Screen-to-Camera (S2C)***

Uma linha de pesquisa ainda mais inédita que utiliza OCC é a de *Screen-to-Camera*. Sistemas S2C se classificam dentro da categoria PHY-VI, pois são utilizados *displays* como transmissores e câmeras como receptores. A captura das imagens das telas pelas câmeras permite o emprego de algoritmos de processamento de imagens, como redes neurais e outros algoritmos de visão computacional, que abrem um novo leque de técnicas de transmissão não presentes em sistemas puramente do tipo VLC ou OCC. (ASHOK et al., 2015; FENG et al.,

2023) Além disso, diferentemente das tecnologias de VLC e OCC, ainda não houve muito estudo sobre a utilização de S2C para posicionamento e localização *indoor*. A Figura 2.3 ilustra o funcionamento básico de um canal S2C com a utilização de um Código QR.

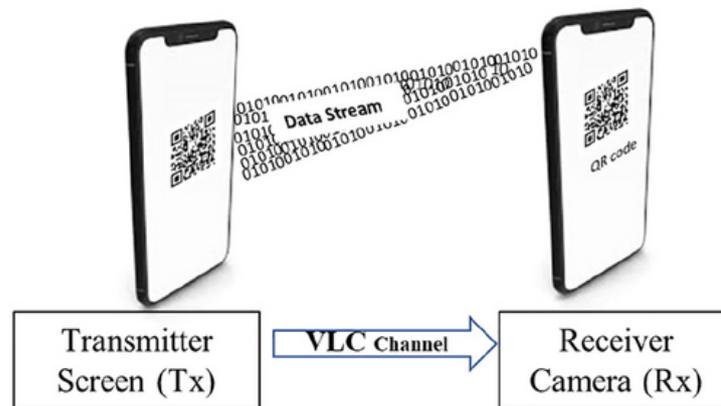


Figura 2.3: Exemplo simplificado de um sistema S2C, sendo a tela de um *smartphone* o transmissor, e a câmera de outro *smartphone* o receptor. (YOKAR et al., 2023)

Além da necessidade de atentar-se aos efeitos de *flickering* nas telas, as modulações empregadas em sistemas S2C provocam distorções no brilho e/ou cores das imagens transmitidas, portanto é um obstáculo a mais para ser mitigado, pois a função primária dos *displays* é transmitir de maneira satisfatória imagens e vídeos para o ser humano.

Felizmente para sistemas S2C, telas digitais são compostos por grandes quantidades de pixels posicionados um imediatamente ao lado do outro. Essa geometria possibilita a caracterização de um sistema de comunicação *Multiple-Input and Multiple-Output* (MIMO) ou *Multiple-Input and Single-Output* (MISO) massivo, que são sistemas comumente encontrados em pesquisas de VLC/OCC (SEJAN et al., 2023), e recentemente definidos pela revisão de 2024 do IEEE 802.15.7 (IEEE..., 2024). Por exemplo, cada pixel da tela poderia ser considerado um elemento transmissor. A redundância dos sinais transmitidos em sistemas MIMO aumenta a robustez da informação propagada pelo meio até atingir o receptor, podendo minimizar perdas por degradações do canal e falhas isoladas de alguns dos transmissores, como pixels queimados ou travados.

## 2.4 Visible Light Positioning

A principal tecnologia para localização e posicionamento externo é o Sistema de Posicionamento Global - GPS. O GPS funciona através de uma malha de 24 satélites que completam uma órbita ao redor da Terra a cada 12 horas e enviam constantemente

informações sobre suas coordenadas através de ondas de RF (KAPLAN; HEGARTY, 2006). Tais ondas são transmitidas em frequências de 1575,42 MHz, 1227,6 MHz e 1176 MHz que são captadas por receptores capazes de descobrir sua posição em latitude, longitude e altitude a partir da comparação do tempo de recepção dos sinais de ao menos quatro satélites. (WING; EKLUND; KELLOGG, 2005)

Quando em ambientes externos, com pouca variação de relevo e obstáculos, o GPS comercial comum é capaz de atingir precisões de até poucos metros, porém quanto maior essa quantidade de obstáculos, e principalmente em ambientes *indoor*, menos confiável fica a utilização desse sistema, pois as ondas RF sofrem muita degradação e interferências de multipercurso (MERRY; BETTINGER, 2019). Um GPS de precisão *Real Time Kinematic* (RTK) é capaz de atingir níveis de precisão de poucos centímetros (MONICO, 2008), porém ainda é susceptível às interferências e degradações da tecnologia RF quando em ambientes *indoor*, além de ser um tipo de equipamento caro para o uso civil. Aqui surge o potencial para explorar formas de posicionamento com luz visível para ambientes internos.

Encontra-se na literatura estudos que utilizaram lâmpadas LED como transmissores e PDs como receptores para sistemas de posicionamento e localização *indoor*. O uso desses materiais justifica-se pelo baixo custo de implementação, facilidade de instalação e manipulação, sendo elementos não invasivos aos demais equipamentos eletrônicos. (LIN, B. et al., 2017; ZHUANG et al., 2018)

Uma etapa crucial para a implementação de alguns desses sistemas é a modelagem do canal de comunicação. Essa modelagem pode ser feita através de alguns métodos, sendo que um dos mais aplicados é a Resposta ao Impulso do Canal - *Channel Impulse Response* (CIR). Essa resposta leva em conta a Potência do sinal Recebido - *Received Signal Strength* (RSS) sob LoS, influência de multipercursos, ângulo de partida (*angle of departure*) do Tx e ângulo de incidência (*angle of arrival*) sobre o fotodetector. (RODRIGUES; XIMENES, 2020; VATANSEVER; LIAN; BRANDT-PEARCE, 2019)

Para a determinação da posição de um Receptor (Rx) dentro do ambiente, dois métodos muito usados são o de trilateração e o de triangulação. Ambos utilizam as informações de RSSs e/ou CIR para cálculo de posicionamento. Na trilateração também é usado o tempo de transmissão entre cada enlace para cálculo da distância do Rx em relação a cada Tx. Já na triangulação, também são utilizados os ângulos de incidência dos sinais captados sobre o sensor

PD para determinação de sua posição relativa a cada Tx. (VATANSEVER; LIAN; BRANDT-PEARCE, 2019)

A maneira mais comum de realizar a trilateração é utilizando três ou mais transmissores - multilateração - como demonstrado na Figura 2.4, para capturar a intersecção dos sinais transmitidos. Entretanto, há estudos que aplicaram técnicas inovadoras e redes neurais para obter resultados similares de posicionamento com dois ou até mesmo um LED transmissor. (LIN, X.; ZHANG, L., 2020)

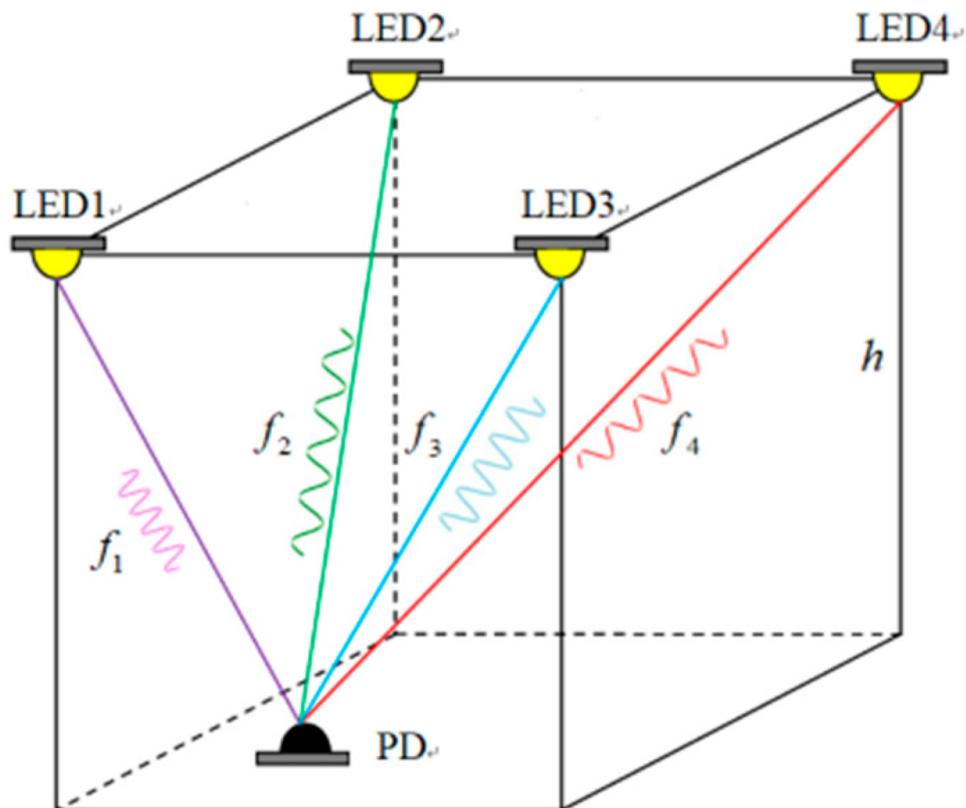


Figura 2.4: Exemplo de um sistema VLP que utiliza lâmpadas de LED no teto em diferentes comprimentos de onda como referências para que o receptor consiga realizar um processo de trilateração. Os diferentes comprimentos de onda garantem que não ocorrem interferências entre os sinais de cada transmissor. (ZHANG, H. et al., 2019)

Outro método de posicionamento é o conhecido como *Fingerprinting*. Essa técnica funciona através de duas etapas, onde em primeiro momento é realizada a coleta de CIRs na maior quantidade possível de coordenadas dentro do ambiente. Essas coletas são armazenadas em um banco de dados e formam o conjunto de dados do *fingerprint* do espaço. Posteriormente, os equipamentos que realizarão a localização através da VLC terão em memória esse banco de dados para comparar em tempo real com os CIR processados conforme sua movimentação, determinando sua posição através dessa comparação. (PATHAK et al., 2015)

## 2.5 Imagens digitais

Uma imagem pode ser armazenada digitalmente através de duas maneiras: matriz de pixels (imagens *raster*) e vetores (imagens vetoriais) (TELEDYNE PHOTOMETRICS, s/d; STS, s/d). Para trabalhos com foco em sistemas OCC e S2C, são usadas imagens *raster*. Como exemplo na Figura 2.5: no padrão RGB, cada imagem colorida é composta pela combinação de 3 camadas de matrizes de dimensões iguais, e possivelmente mais uma camada para um canal *alpha* transparente, dependendo do formato original da imagem, como no caso de um png. Cada matriz armazena os valores dos pixels que compõem a imagem referentes à cor daquela camada. Se determinado pixel apresenta os valores (0,0,255) no padrão RGB sem canal *alpha*, sendo 0 ausência de cor, e 255 a intensidade máxima da camada, sua cor é azul. (MASON, 1993; MEER; GEORGESCU, 2001)

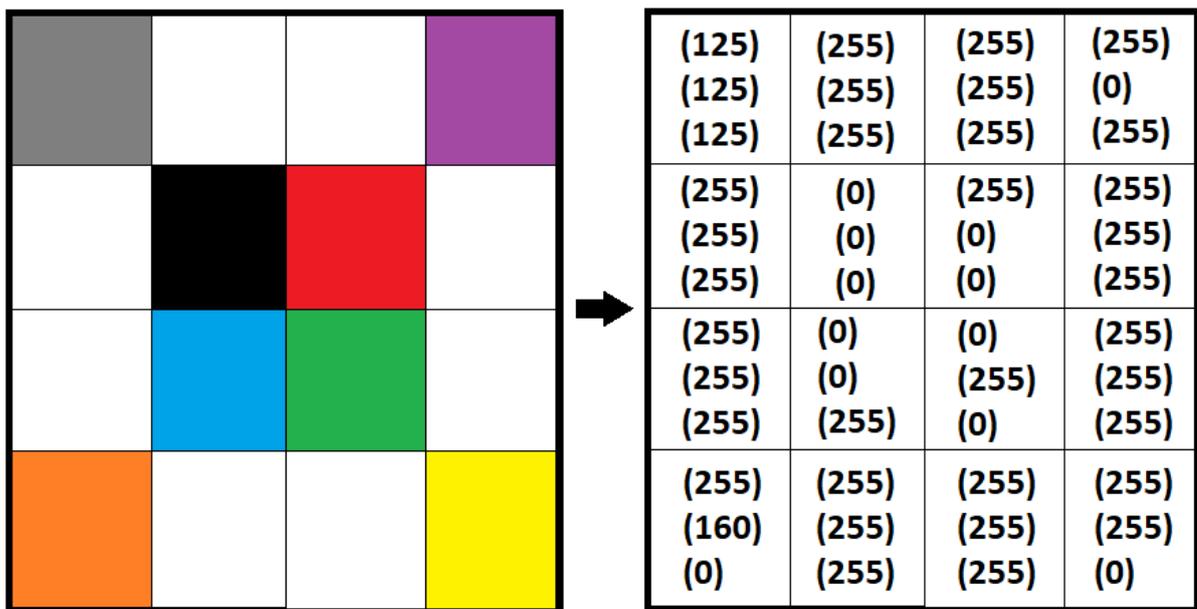


Figura 2.5: Exemplo simplificado da formação de cores pelo padrão RGB sem canal *alpha*. Para formar o quadriculado colorido do retângulo à esquerda, os valores das intensidades dos pixels de cada camada vermelha, verde e azul estão representados no quadrado à direita.

Junto com o padrão RGB, os principais sistemas de cores incluem: (GONZALEZ; WOODS, 2010a; SINGH, 2023)

- Escala de cinza: Representação em apenas uma camada onde o nível de intensidade do pixel é convertido em uma escala de cinzas, entre 0 (preto) e o limite de armazenamento, como por exemplo, 255 para 1 pixel armazenado em 8 bits.

- *Red, Green, Blue* (RGB): Utiliza a combinação de luz vermelha, verde e azul para criar cores, sendo amplamente utilizado em dispositivos eletrônicos e telas.
- *Cyan, Magenta, Yellow, Black* (CMYK): Comumente usado em impressão, onde as cores são sobrepostas em camadas para criar uma ampla gama de cores.
- *Hue, Saturation, Lightness/Value* (HSL/HSV): Define as cores com base em matiz, saturação e luminosidade/valor.
- YCbCr: representa os pixels em uma escala de luminância (Y), diferenças de cromaticidade azul (Cb) e vermelho (Cr).

Trabalhar com imagens em escala de cinza ao invés do padrão colorido permite reduzir o *array* de cores para apenas uma camada, onde a intensidade é referente apenas a uma escala de cinza, do preto (valor 0) ao branco (valor 255). Dessa forma reduz-se a complexidade de códigos para processamento de imagens. (ITU-T, 2011)

## 2.6 *Processamento de Imagens*

Para o desenvolvimento deste trabalho, foram utilizados conceitos comumente aplicados em sistemas de visão computacional e processamento de imagens digitais, que vão desde a definição de um padrão colorido (Seção 2.5) e em escala de cinzas de uma imagem digital, às técnicas clássicas de processamento de imagens mais utilizadas para otimizar o desempenho de algoritmos de identificação de objetos, os principais métodos de visão computacional para transformações e cálculos de perspectivas, como também definições pertinentes à criação de protocolos de comunicação visual.

Desenvolver um sistema S2C significa encontrar formas de transmitir dados utilizando como transmissor uma ou diversas telas digitais, e receber tais dados por uma ou diversas câmeras como sensores de imagens. A propagação da mensagem codificada é feita de forma espacial com a variação de intensidades dos pixels, formando diferentes símbolos conforme a modulação empregada, e também de forma temporal, com a segmentação das informações transmitidas ao longo dos quadros de vídeo. (SESYUK; IOANNOU; RASPOPOULOS, 2022)

Uma ou diversas câmeras devem capturar os quadros transmitidos, identificar a mensagem modulada no padrão especificado e decodificá-la. Para realizar a correta identificação dessa mensagem, trabalhar com os quadros capturados em suas condições

originais não produz acurácia suficiente em grande parte dos canais de comunicação em rotinas de detecção de objetos. Para melhorar esse desempenho, são necessários pré processamentos quadro a quadro para destacar ao máximo o tipo de informação que deseja-se capturar.

Na área de Visão Computacional clássica, alguns algoritmos de pré processamento estão presentes em praticamente todos os sistemas que utilizam processamento de imagens e vídeos, muitas vezes em conjunto:

- Conversão para escala de cinza: é um processo no qual uma imagem colorida é convertida em uma representação em tons de cinza, onde cada pixel possui um único valor de intensidade representando o brilho ou a luminosidade. Isso é frequentemente feito para simplificar o processamento de imagens e reduzir a quantidade de dados necessários para armazenar e manipular a imagem. Com uma imagem no padrão RGB, é possível converter as intensidades dessas três camadas com a Equação 2.1 de soma de produtos que atribui pesos a cada camada (*R - Red*, *G - Green*, *B - Blue*), conforme a recomendação ITU-R BT.601 adotada por equipamentos de transmissão nos padrões PAL e NTSC (ITU-T, 2011):

$$P = 0,3R + 0,59G + 0,11B \quad (2.1)$$

- Equalização de histograma: técnica utilizada para melhorar o contraste em uma imagem. (SHANMUGAVADIVU; BALASUBRAMANIAN, 2014) Ela redistribui os valores de intensidade dos pixels de forma a ampliar a faixa de intensidades, espalhando de forma mais homogênea os pixels com intensidades próximas, resultando em uma imagem com maior contraste e detalhes mais visíveis. É tomado como base os valores de intensidade dos pixels de acordo com o histograma da imagem original.
- Limiarização: método de segmentação de imagens que as divide em regiões ou objetos com base em um limite de intensidade de pixels predefinido. Pixels com intensidades acima desse limite são atribuídos a uma classe, enquanto pixels com intensidades abaixo são atribuídos a outra. É uma técnica comum para separar objetos de interesse do restante da imagem. A partir da limiarização, pode-se realizar uma binarização, onde pixels com intensidade acima do limiar são atribuídos ao valor máximo de

intensidade, e os pixels abaixo são completamente apagados. (SHANMUGAVADIVU; BALASUBRAMANIAN, 2014; CHEN, S. D.; RAMLI, 2003)

- Filtragem espacial ou de frequências: técnicas de processamento de imagens para remoção de ruídos de alta frequência para suavizar transições, realce de bordas, destaque de tons intermediários, identificar bordas ou outras características da imagem por meio da aplicação de filtros no domínio espacial através de convolução entre os pixels da imagem e matrizes de filtragem, ou na frequência, com multiplicações entre as transformadas de Fourier da imagem e do filtro. (MASON, 1993)
- Operações morfológicas: técnicas que envolvem a manipulação da forma ou estrutura de objetos em uma imagem. Elas incluem operações como erosão, dilatação, abertura e fechamento, que são usadas para limpar ruídos, separar objetos conectados, unir regiões e extrair características importantes da imagem. Essas operações são baseadas em conjuntos matemáticos e operações sobre conjuntos, e são frequentemente usadas em conjunto com outras técnicas de processamento de imagens para análise e manipulação de objetos em imagens digitais. (MASON, 1993)

Com o pré processamento dos quadros, deseja-se facilitar a etapa de identificação das informações pertinentes das imagens para o sistema de comunicação. Este pós processamento pode também ser realizado com técnicas clássicas, como algoritmos identificação de bordas para estipular regiões de interesse, como o algoritmo de Canny (CANNY, 1986), identificação de padrões pré definidos como marcos de referência, subtração de quadros para isolar as informações que variam na composição das mensagens dos demais elementos estáticos da paisagem, entre outras técnicas. (MEER; GEORGESCU, 2001)

Também vem sendo explorado o uso da Inteligência Artificial para essa etapa de identificação de objetos de interesse. Para tal, é necessário o treinamento de um classificador sobre um *dataset* com diversos elementos e cenários que deseja-se identificar posteriormente. Muitas vezes é interessante trabalhar com ambos os algoritmos tradicionais de pré e pós processamento junto ao modelo a ser treinado, aumentando a acurácia final das identificações. (ROTH, 1990)

## 2.7 Operações Morfológicas

Muitas vezes a perspectiva de um objeto em relação ao Rx, composta pelo ângulo de captura dos quadros, a distância e a resolução da câmera influenciam a nitidez e geometria de determinados pontos no espaço. Se um código de barras é capturado perfeitamente alinhado de forma centralizada horizontalmente e verticalmente, a uma curta distância onde o código ocupe mais de 50% da área do quadro, as chances de um algoritmo identificador de código de barras retornar um resultado positivo são muito maiores do que se esse mesmo código fosse capturado com uma angulação de 60° vertical e uma distância onde ele ocupe 5% do quadro. Nessas condições, dependendo da resolução e densidade de pixels da câmera, as linhas do código sofrerão serrilhados em suas bordas e efeitos de sobreposição entre elas.

Aplicar operações morfológicas em uma imagem significa utilizar um elemento estruturante que pode ser uma forma geométrica, como linhas, quadrados, elipses, entre outras formas. para alterar as propriedades de determinadas regiões, como deixar traços mais grossos ou conectar pontilhados. (GONZALEZ; WOODS, 2010b) Considerando uma imagem com um elemento visual  $A$  e um elemento estruturante  $B$ , as principais operações morfológicas utilizadas para processamento de imagens são explicadas a seguir, e demonstradas na Figura 2.6:

- Erosão: consiste em percorrer um elemento estruturante  $B$  ao longo dos pixels que beiram todo o contorno do elemento original  $A$ , e realizar uma subtração onde ocorre intersecções entre os pixels do elemento original e o estruturante, como na Equação 2.2.

$$(A \ominus B) = \bigcap_{b \in B} (A)_{-b} \quad (2.2)$$

- Dilatação: de certa forma é o processo inverso da erosão. Na dilatação, deseja-se expandir os contornos da forma original percorrendo o elemento estruturante internamente aos pixels do elemento original, e unindo os valores dos pixels diretamente externos ao contorno original, como na Equação 2.3.

$$(A \oplus B) = \bigcup_{b \in B} (A)_b \quad (2.3)$$

- Abertura: é uma combinação do processo de dilatação aplicado após um processo de erosão, podendo ser o mesmo elemento estruturante em ambas as etapas, ou elementos

diferentes, como na Equação 2.4. Geralmente usado para suavização de contornos e remoção de ruídos:

$$(A \circ B) = (A \ominus B_1) \oplus B_2 \quad (2.4)$$

- **Fechamento:** Oposto ao processo de abertura, envolve primeiramente a operação de dilatação, e posteriormente a de erosão. Novamente, ambas as etapas podem usar o mesmo elemento estruturante ou elementos diferentes, como na Equação 2.5. Geralmente usada para conectar partes possivelmente distorcidas e desconexas.

$$(A \bullet B) = (A \oplus B_1) \ominus B_2 \quad (2.5)$$

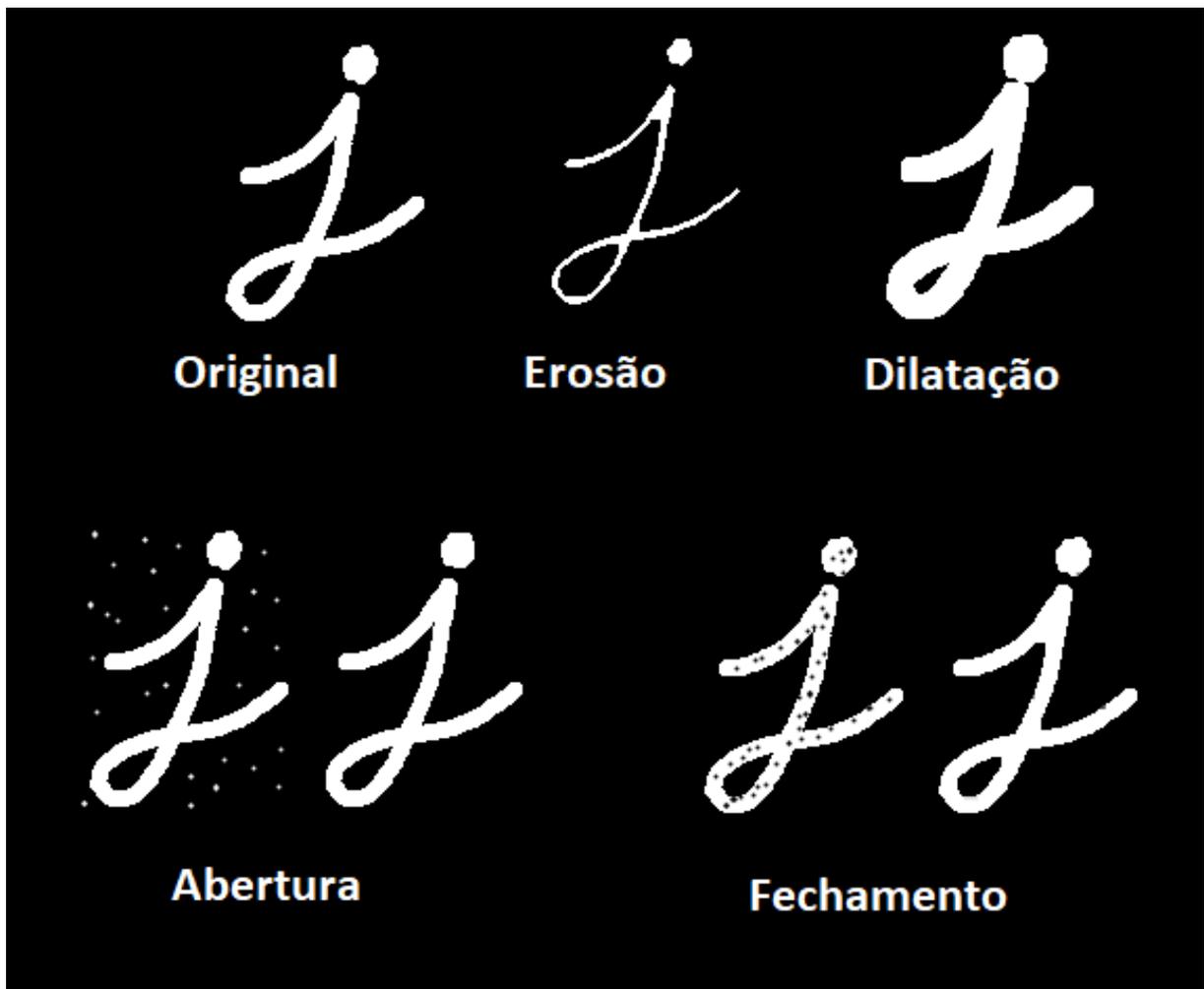


Figura 2.6: Exemplos das principais operações morfológicas. Adaptado de (OPENCV CONTRIBUTORS, s.d.)

## 2.8 Detector de Bordas de Canny

O algoritmo de Canny é um dos métodos mais populares para detecção de bordas em imagens (CANNY, 1986). Sua primeira etapa envolve a suavização da imagem para reduzir o ruído e evitar que detalhes irrelevantes sejam interpretados como bordas. Para isso, utiliza-se o filtro Gaussiano (HADDAD; AKANSU, 1991; SHAPIRO; STOCKMAN, 2001) conforme a Equação 2.6:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.6)$$

onde  $\sigma$  é o desvio padrão da função Gaussiana. É então feita uma operação de convolução entre a imagem de entrada  $I(x, y)$  e  $G(x, y)$ :

$$I_{\text{suavizada}}(x, y) = I(x, y) * G(x, y) \quad (2.7)$$

Após a suavização, o gradiente de intensidade da imagem é calculado utilizando derivadas no eixo  $x$  e  $y$ . Para isso, são utilizados operadores como Sobel (SOBEL, 2014):

$$G_x = \frac{\partial I_{\text{suavizada}}}{\partial x}, \quad G_y = \frac{\partial I_{\text{suavizada}}}{\partial y} \quad (2.8)$$

A magnitude do gradiente  $G$  e sua direção  $\theta$  são obtidas como:

$$G = \sqrt{G_x^2 + G_y^2}, \quad \theta = \arctan\left(\frac{G_y}{G_x}\right) \quad (2.9)$$

Para afinar as bordas detectadas, realiza-se a supressão de não-máximos, onde o gradiente é analisado na direção  $\theta$ , e apenas os pixels que representam máximos locais são mantidos:

$$G(x, y) = \begin{cases} G(x, y) & \text{se } G(x, y) \text{ for máximo local} \\ 0 & \text{caso contrário} \end{cases} \quad (2.10)$$

Nesta etapa, utiliza-se dois limiares,  $T_{\text{alto}}$  e  $T_{\text{baixo}}$ , para identificar as bordas fortes e fracas:

- Pixels com  $G(x, y) > T_{\text{alto}}$  são classificados como bordas fortes. - Pixels com  $T_{\text{baixo}} < G(x, y) \leq T_{\text{alto}}$  são classificados como bordas fracas, sendo mantidos apenas se estiverem conectados a bordas fortes. - Pixels com  $G(x, y) \leq T_{\text{baixo}}$  são descartados.

A Figura 2.7 demonstra a aplicação do Detector de Bordas de Canny sobre a clássica imagem 'Cameraman', comumente usada no mundo científico em testes de processamento de imagens.

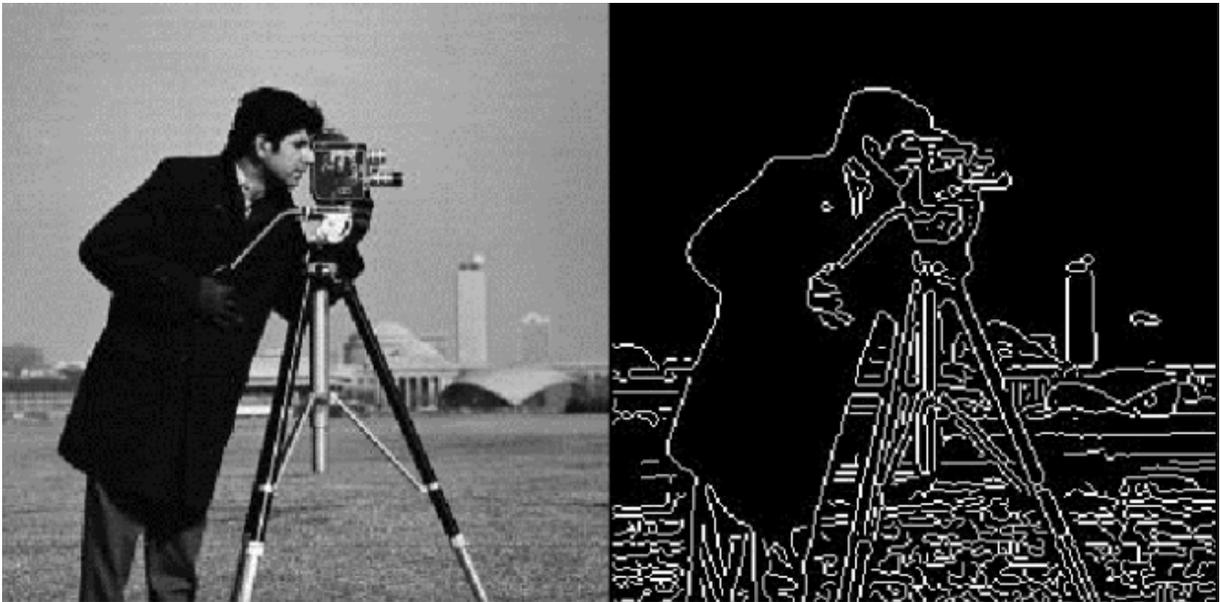


Figura 2.7: Exemplo do Detector de Bordas de Canny sobre a imagem 'Cameraman'.

## 2.9 Homografia e perspectivas de imagens

A Homografia é uma transformação geométrica fundamental em visão computacional e processamento de imagem, especialmente em aplicações como calibração de câmera e registro de imagens. A transformação por homografia é usada para mapear pontos das coordenadas do espaço real para as coordenadas digitais da câmera. (LU, 2018; OPENCV CONTRIBUTORS, n.d.)

Uma matriz de homografia é geralmente definida como:

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \quad (2.11)$$

Para um par de pontos correspondentes  $(x, y)$  e  $(u, v)$ , onde  $(x, y)$  é o ponto no plano de imagem original e  $(u, v)$  é o ponto transformado, a relação é dada pela equação 2.12:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2.12)$$

Essa Equação pode ser simplificada para a Equação 2.13:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11}x + h_{12}y + h_{13} \\ h_{21}x + h_{22}y + h_{23} \\ h_{31}x + h_{32}y + h_{33} \end{bmatrix} \quad (2.13)$$

A matriz de homografia  $H$  é uma combinação linear das matrizes de transformação geométrica básicas, como escalonamento  $S$ , rotação  $R$ , translação  $T$  e cisalhamento  $C$ . (GONZALEZ; WOODS, 2010c) As matrizes de transformação básicas geralmente são representadas conforme as Equações 2.14, 2.15, 2.16 e 2.17:

$$\mathbf{S} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.14)$$

$$\mathbf{R} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.15)$$

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.16)$$

$$\mathbf{C} = \begin{bmatrix} 1 & h_x & 0 \\ h_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.17)$$

Sendo assim, a matriz de homografia  $H$  pode ser expressa como uma composição dessas transformações básicas na Equação 2.18:

$$H = S \cdot R \cdot T \cdot C \quad (2.18)$$

A Figura 2.8 demonstra de forma ilustrativa tais operações de homografia. O processo de homografia para transformação de quadros de vídeo foi aplicado integralmente no editor de perspectivas apresentado no Apêndice (A), desenvolvido para auxiliar a pesquisa ao longo de simulações e testes reais.

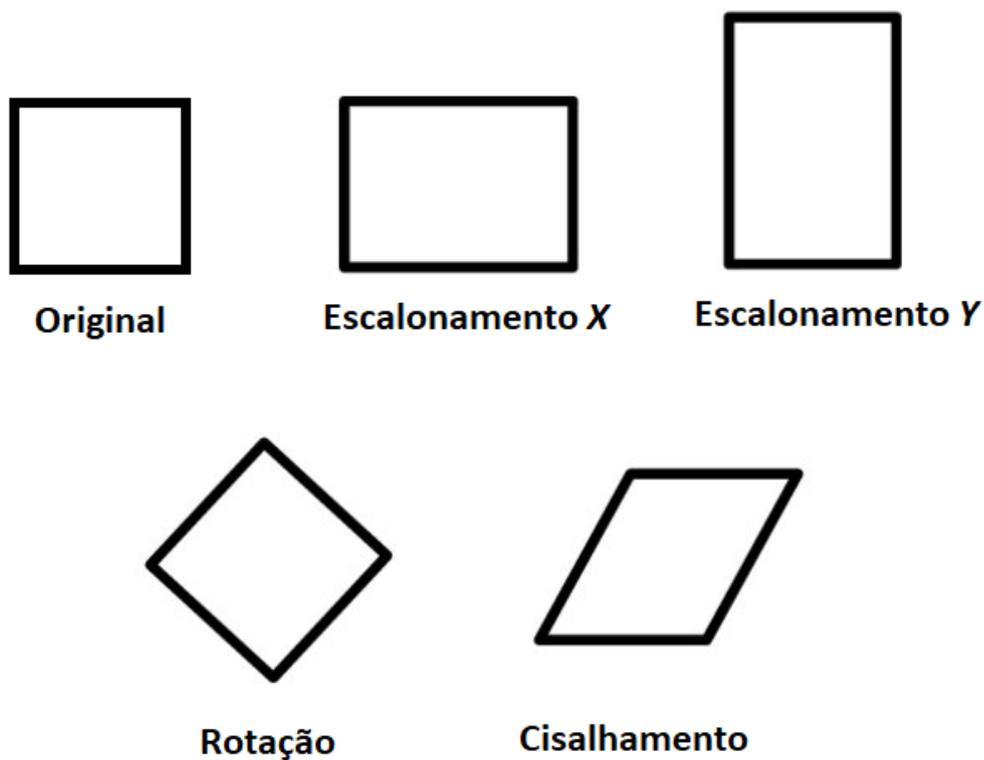


Figura 2.8: Exemplos isolados de Homografia, tomando um quadrado como forma original. Adaptado de (DREWES; PFEUFFER; ALT, 2019)

## 2.10 Código de Barras

O Código de Barras 128 (BC-128) foi desenvolvido pela Computer Identics Corporation em 1981. Trata-se de um código capaz de representar todos os 128 caracteres do padrão ASCII,

garante grande flexibilidade para codificar dados alfanuméricos digitados por um teclado de computador tradicional e pode ser usado por sistemas que empregam processamento de imagens ou visão computacional para transmissão e leitura de dados codificados. O BC-128 demonstrado na Figura 2.9 funciona da seguinte forma (KEYENCE CORPORATION, s.d.):

- Existem quatro tipos de tamanhos de barra no BC-128.
- Cada caractere é composto por três barras e três espaços, totalizando seis elementos.
- O caractere de início possui três opções: *A*, *B* e *C*.
  - A escolha do caractere de início define quais caracteres podem ser representados posteriormente.
  - Por exemplo, ao usar *A* como caractere de início, pode-se representar os caracteres listados na coluna correspondente a *A* na tabela de referência do código em (EXPRESS CORPORATION, s.d.).
- Quando se utiliza *C*, números de dois dígitos podem ser representados por um único padrão de barra.
- Durante o processamento, é possível alternar entre os conjuntos de caracteres (*A*, *B* e *C*).
  - Por exemplo, um código de barras iniciado com *A* pode ser alterado para usar caracteres de *B* ou *C*.
- O comando *SHIFT* permite alternar temporariamente para o conjunto de caracteres seguinte (*A* para *B*, *B* para *C* ou *C* para *A*).
  - Essa alteração é válida apenas para o caractere imediatamente após o *SHIFT*, funcionando de maneira semelhante à tecla *SHIFT* em um teclado de computador.
- O dígito de verificação do código de barras é calculado utilizando o **Módulo 103**.



Figura 2.9: Exemplo de um Código de Barras 128. Adaptado de (KEYENCE CORPORATION, s.d.)

A tabela de caracteres para configuração de um Código de Barras 128 encontra-se no Anexo (7.4)

## 2.11 *Perspective-n-Point*

Em sistemas de detecção de objetos ou de posicionamento utilizando uma câmera receptora, muitas vezes é desejado estimar a pose que um objeto foi observado pelo sensor da câmera. Um dos métodos mais conhecidos e validados para realizar tal estimação é o *Perspective-n-Point* (PnP) (QUAN; LAN, 1999).

O PnP envolve estimar as transformações de rotação e translação que melhor se ajustam à projeção dos pontos 2D da imagem  $(u_i, v_i) \in I$  de cada ponto do objeto de interesse sobre os seus pontos 3D correspondentes  $(x_i, y_i, z_i)$  no espaço (FAUGERAS, 1993; WONG; MENDONCA; CIPOLLA, 2003). A Figura 2.10 ilustra um mapeamento imagem-objeto (ou virtual-real) das coordenadas de uma tela, tomando oito pontos de referência.

A solução para o PnP envolve determinar a matriz de rotação  $\mathbf{R} \in \mathbb{R}^{3 \times 3}$  e o vetor de translação  $\mathbf{t} \in \mathbb{R}^3$  que satisfazem a equação de projeção a seguir:

$$\mathbf{p}_i = \mathbf{K}(\mathbf{R}\mathbf{P}_i + \mathbf{t}), \quad \forall i = 1, 2, \dots, n$$

- $\mathbf{P}_i = \begin{bmatrix} X_i & Y_i & Z_i \end{bmatrix}^\top$  são as coordenadas dos pontos 3D no sistema de referência global.
- $\mathbf{p}_i = \begin{bmatrix} u_i & v_i \end{bmatrix}^\top$  são as coordenadas projetadas no plano da imagem em pixels.

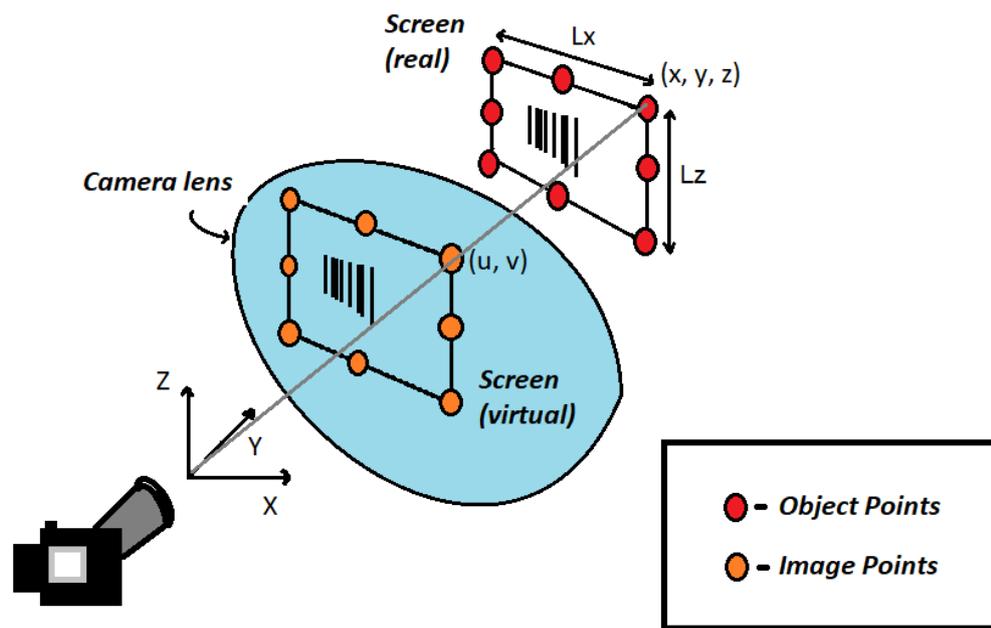


Figura 2.10: Problema de Perspectiva- $n$ -Ponto para estimativa de pose.

- $\mathbf{K}$  é a matriz intrínseca da câmera, definida por:

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix},$$

onde  $f_x$  e  $f_y$  são as distâncias focais (em pixels) e  $(c_x, c_y)$  é o centro óptico da câmera.

Para o caso particular em que  $n = 3$ , ou seja, três correspondências ponto a ponto são fornecidas, o problema pode ser resolvido de forma única sob as seguintes condições:

- Os três pontos 3D  $\mathbf{P}_1$ ,  $\mathbf{P}_2$  e  $\mathbf{P}_3$  não são colineares, ou seja, definem um plano e possuem coordenadas 3D distintas.
- A matriz intrínseca  $\mathbf{K}$  da câmera é conhecida e bem calibrada, de modo que as projeções possam ser corretamente relacionadas com as coordenadas do espaço tridimensional.

## 2.12 Breve Revisão Literária

Para o levantamento bibliográfico deste trabalho, foi realizada primeiramente uma busca por outros trabalhos contextualizados dentro das vertentes de VLC, OCC e S2C para

contextualização do atual estado da arte. Conforme a necessidade de reduzir o escopo de busca dentro dos artigos, publicações em periódicos, seminários, jornais, entre outros, com maior relação ao tema de comunicação S2C para posicionamento e transmissão de dados, foram ajustados os parâmetros de busca para manter em alerta novidades durante o desenvolvimento desta pesquisa.

O levantamento está dividido nas seções à seguir com uma quantificação dos resultados retornados na revisão literária sistêmica, seguido de uma breve contextualização das principais linhas de pesquisa, e um maior aprofundamento dos conceitos e estudos pertinentes às pesquisas, padronizações e tecnologias já existentes para VLC, OCC e S2C.

Uma *string* de busca foi proposta para evidenciar apenas os trabalhos com linha de pesquisa em posicionamento *indoor* com OCC e que contenham o uso de telas de alguma forma. Tais buscas foram realizadas nas bases do IEEEEXPLORE, Scopus e Web of Science:

**(ALL ("Visible Light Positioning") OR ALL ("Optical Camera Communication") AND ALL ("Screen")) AND (ALL ("Indoor") OR ALL ("Color Shift Keying"))**

Foi realizado o processo de estudo e revisão desses artigos e também a constante atualização com eventuais publicações que acionem os alertas configurados nos *websites*. Também foram realizadas buscas manuais para complementar as referências com base nesse banco de dados e outros, como o ScienceDirect e o MDPI. Foi encontrada uma pequena quantidade de trabalhos relacionados à área com um caráter de maior foco no posicionamento *indoor* de câmeras com processamento de códigos de referência transmitidos, e até então, nenhum estudo aborda diretamente a criação de um sistema S2C para posicionamento da maneira como proposto neste trabalho, tornando ainda mais interessante esta pesquisa como potencial inovação.

Para compreender as necessidades e motivações que estimularam esta linha de pesquisa, é preciso primeiro contextualizar brevemente o atual estado da arte das tecnologias de comunicação e posicionamento indoor (MENDOZA-SILVA; TORRES-SOSPEDRA; HUERTA, 2019). Dentre os maiores desafios e demandas desta área, destacam-se a necessidade de precisão e exatidão em níveis de poucos centímetros para o sistema de posicionamento (visto que ambientes *indoor* geralmente possuem grande densidade de obstáculos em um espaço físico limitado), a utilização de novas faixas do espectro eletromagnético que não conflitem com outras tecnologias já consolidadas, e explorar a iluminação e equipamentos que emitem sinais luminosos já amplamente presentes nos ambientes construídos pelo ser humano para

agregar novas utilidades com o menor impacto em infraestrutura e custo de implementação. (O'BRIEN et al., 2008)

Diversos trabalhos já foram propostos com tecnologias emergentes para posicionamento e localização indoor visando alta exatidão e precisão de poucos centímetros. Alguns estudos focam na utilização de Wi-fi, mantendo-se com a natureza de radiofrequência, enquanto outros exploram o Bluetooth Low Energy como alternativa, conforme revisões seminais como em (SESYUK; IOANNOU; RASPOPOULOS, 2022). Há também o desenvolvimento de sistemas híbridos, onde o foco é suprir com um tipo de tecnologia as limitações da outra para tornar a obra mais robusta como um todo. (GUAN et al., 2021; SUNG; JUNG, S.-h.; HAN, 2015)

Dentro da linha de pesquisa de posicionamento com OCC, o trabalho de (FENG et al., 2023) realiza o posicionamento OCC com uma lâmpada de LEDs retangular onde um código de barras é gravado fisicamente na lâmpada com os dados de identificação estáticos. Este trabalho apresenta propostas similares para processamento do código e identificação dos ângulos da câmera em relação à lâmpada para posicionamento, porém fica limitado a essa gravura na lâmpada, sendo necessário substituir completamente o dispositivo caso seja necessário alterar a mensagem. Com um sistema S2C, além da vantagem de alterar facilmente o conteúdo transmitido na tela, é possível enviar outros tipos de informações para integrar com outras aplicações de comunicação.

Há trabalhos que exploraram a utilização do S2C para posicionamento, porém de uma forma menos flexível. Em (KIM et al., 2021), os pesquisadores empregaram a identificação de um Código QR para realizar uma trilateração com os cantos de referência inerentes do código. Dessa forma, aplicam algoritmos de calibração de câmera e identificação de perspectiva similares aos métodos apresentados no Capítulo 3 e se aproveitam da presença de códigos QR comumente encontrados em telas e *outdoors*, porém não exploraram a dimensão temporal para transmitir os dados, mantendo somente um código fixo como *landmark*, além de não utilizar a informação do código em si.

Encontra-se a mesma linha de aplicação nos trabalhos de (PAN et al., 2020) e (KANG; CHOI; JUNG, J.-W., 2022), onde novamente utilizam as dimensões dos códigos QR fixos para realizar uma trilateração e identificação de perspectiva, porém não se aproveitam do conteúdo do código ou de transmissão de diferentes códigos.

## Capítulo 3

# Desenvolvimento - S2CP com códigos de barras

Para o funcionamento do sistema de posicionamento *indoor* através da comunicação S2C, batizado aqui como *Screen-to-Camera-Positioning* - (S2CP), o objetivo é determinar a posição de uma câmera receptora, seja de um *drone* ou *smartphone* dentro de um cômodo baseando-se na detecção de apenas uma tela, no processamento das mensagens espaço-temporais transmitidas sobre dimensões da tela, e futuramente com demais informações transmitidas sobre a geometria do ambiente e outros dados relevantes de comunicação, como ilustrado na Figura 3.1.

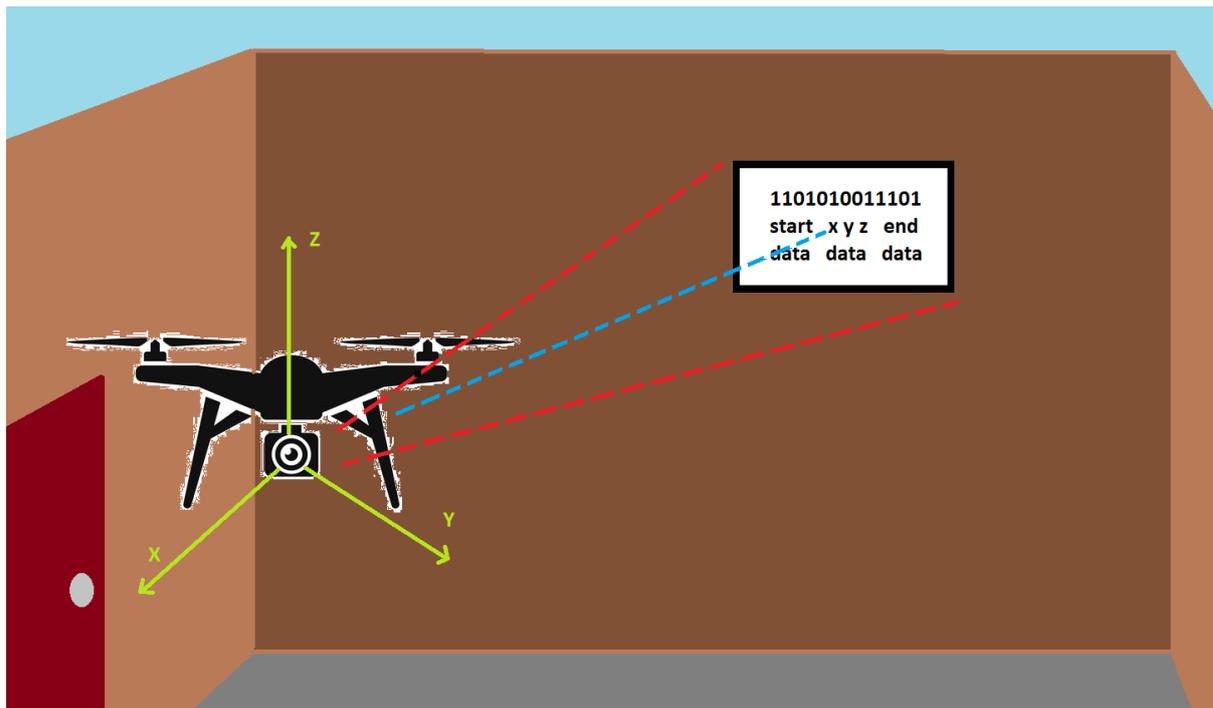


Figura 3.1: Ilustração do sistema proposto onde a câmera de um drone identifica uma tela e processa os dados transmitidos para posicionamento

Muitas vezes a imagem original capturada não está de uma maneira favorável para a extração e processamento de dados de posicionamento e comunicação. Ajustes de saturação, brilho, paleta de cores, borramento, transparência, binarização, entre outros, são necessários para reduzir a complexidade e aumentar a assertividade de algoritmos desenvolvidos para o processamento a ser aplicado.

É possível dividir o sistema de uma forma geral nos seguintes blocos:

- Pré calibração da câmera: esta etapa é um processamento independente do restante do sistema e realizada isoladamente antes de qualquer outro processo.
- Sistema de posicionamento: O algoritmo de posicionamento principal envolve a leitura de quadros de vídeo, identificação da tela e estimativa da posição da câmera. A identificação da tela (ID e dimensões) é obtida por meio de protocolos de comunicação, permitindo flexibilidade para reposicionamento ou substituição da tela.
- Sistema de comunicação: A principal inovação do nosso sistema S2CP é a transmissão de dados que inclui informações sobre as dimensões da tela (para ajustes de posicionamento), bem como a possibilidade de outros tipos de dados, como o identificador único da tela, informações para sincronização de quadros e outros dados

úteis, como telemetria, multimídia ou informações de propósito geral. É aqui que a funcionalidade S2C do nosso método entra em ação.

O sistema completo está ilustrado na Figura 3.2 e pode ser dividido nos seguintes blocos:

- 0: calibração da câmera (uma etapa prévia realizada apenas uma vez);
- 1: embarcar a mensagem codificada no vídeo;
- 2: transmitir o vídeo através de uma tela digital;
- 3: aquisição de vídeo com a câmera;
- 4: pré-processamento dos quadros;
- 5: identificação da tela;
- 6: decodificação da mensagem;
- 7: estimativa das coordenadas.

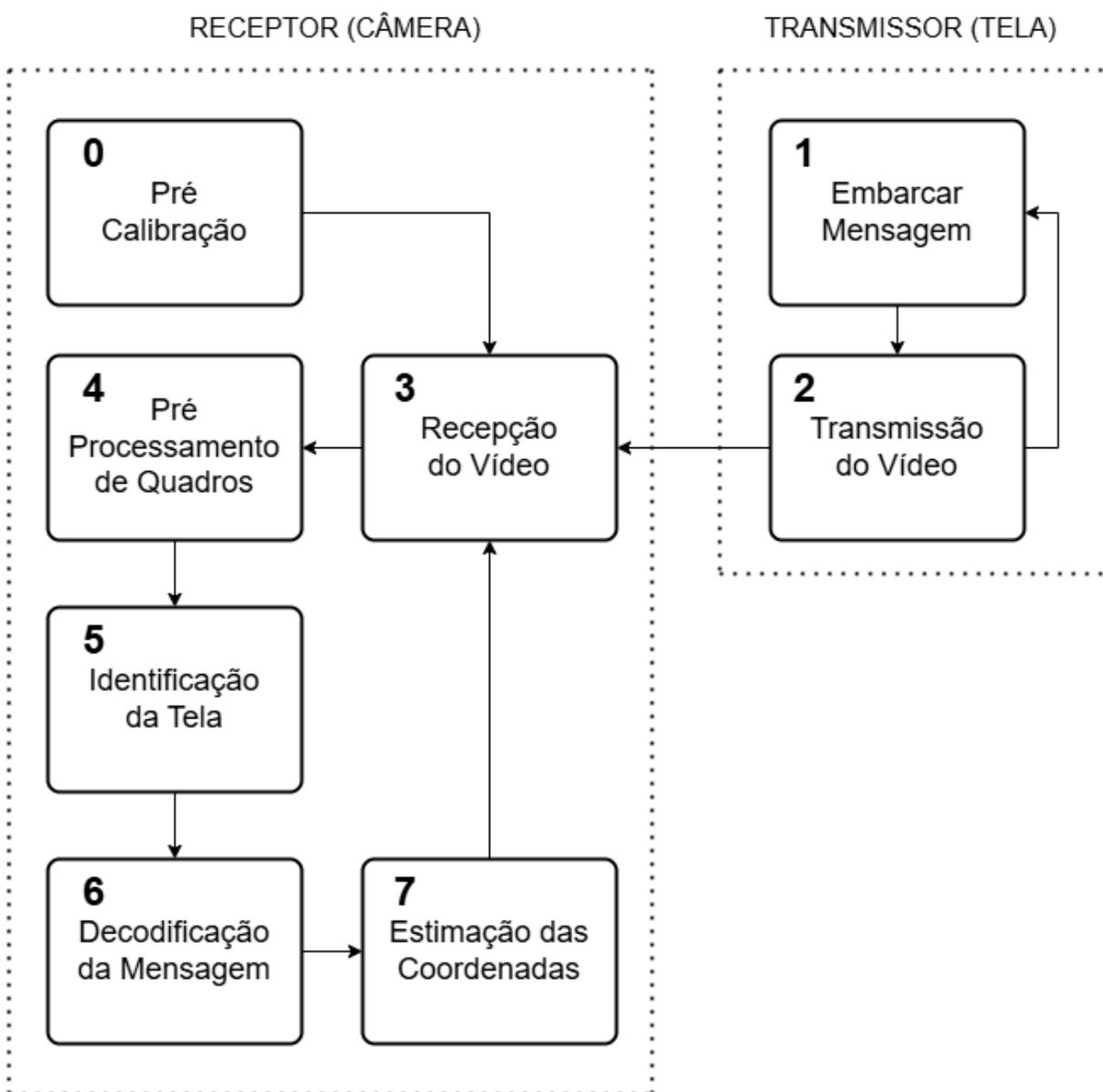


Figura 3.2: Fluxograma do funcionamento completo do sistema desenvolvido. O sistema entra em um regime cíclico de processamento entre os blocos 3 e 7, até finalizar completamente a decodificação de uma mensagem.

Nas seções à seguir, são detalhados os procedimentos e passo a passo para a calibração da câmera, processamento dos quadros, identificação e decodificação das mensagens, cálculo da distância e das coordenadas da câmera em relação ao monitor. Cada etapa será apresentada com fluxogramas para melhor segmentação dos blocos apresentados na Figura 3.2.

### 3.1 Transmissão do Vídeo Codificado

O sistema S2CP proposto utiliza uma única tela de painel plano para exibir dados relacionados às dimensões físicas da mesma, permitindo adaptabilidade a diferentes tamanhos físicos. As informações sobre a altura e largura do *display* são transmitidas por meio de uma sequência de códigos de barras no padrão BC-128. Conseqüentemente, a mensagem deve ser adaptada às dimensões específicas de cada equipamento.

A seqüência de quadros segue o padrão:

1. Uma seqüência de um ou mais quadros com a mensagem *start* embutida é transmitida. Esta mensagem serve como um quadro de sincronia para as informações restantes da tela, bem como para fins de sincronização. Observe que esta mensagem arbitrária não tem efeito perceptível sobre a precisão de posicionamento do S2CP e pode ser substituída por qualquer outra mensagem reservada;
2. Uma seqüência de um ou mais quadros com a mensagem  $(L_x, L_z)$  é transmitida, onde  $L_x$  é a dimensão horizontal da tela e  $L_z$  é a dimensão vertical, ambas em milímetros (Figura 3.1);
3. A mensagem *end* é transmitida para sinalizar o fim do padrão de quadros.

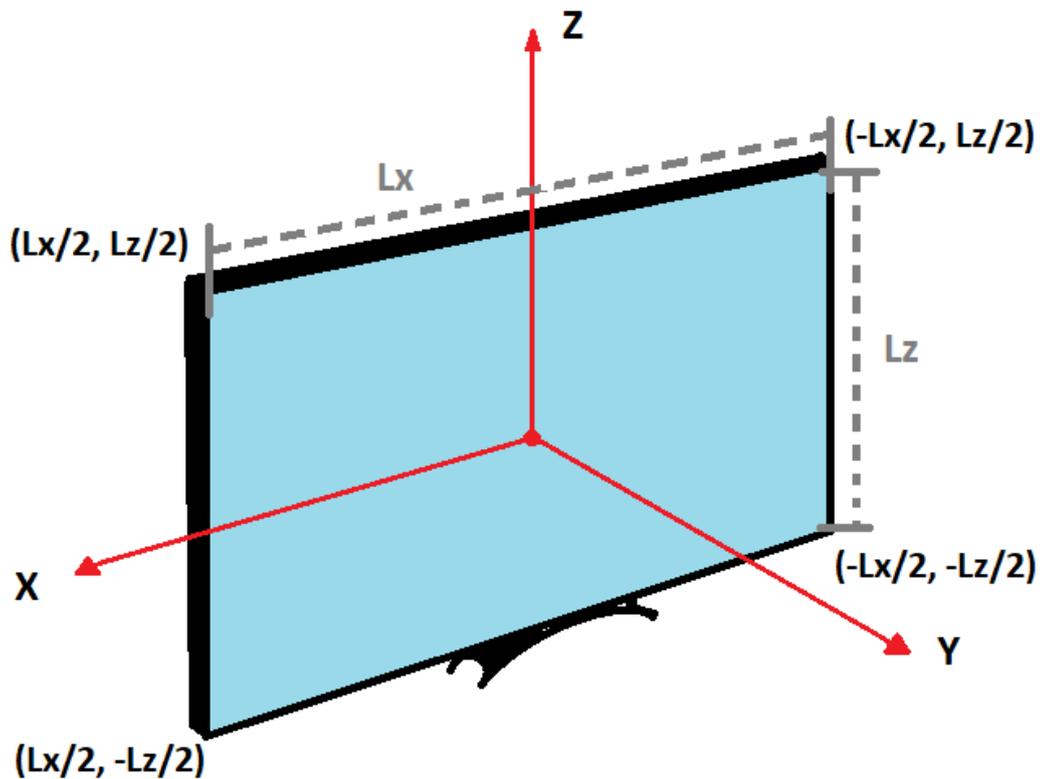


Figura 3.3: Os eixos  $X$ ,  $Y$  e  $Z$  do sistema de coordenadas utilizado, com o centro da tela como a origem. As dimensões  $L_x$  e  $L_z$  da tela são medidas ao longo das linhas cinzas.

Essa sequência de quadros deve ser embutida e enviada periodicamente, como mostrado nos Blocos 1 e 2 da Figura 3.2, para que o receptor possa atualizar as informações de posicionamento, se necessário. Um exemplo desse padrão cíclico de quadros é mostrado na Figura 3.4, onde (565, 330) indica as dimensões ( $L_x, L_z$ ) desta tela arbitrária. Observe que essa escolha arbitrária de transmitir as dimensões físicas, atualmente usando valores numéricos gravados visualmente, poderia ser facilmente substituída por qualquer outro padrão, como um código QR, ou utilizando outro protocolo S2C, como os padronizados em (IEEE STANDARD ASSOCIATION, 2011).

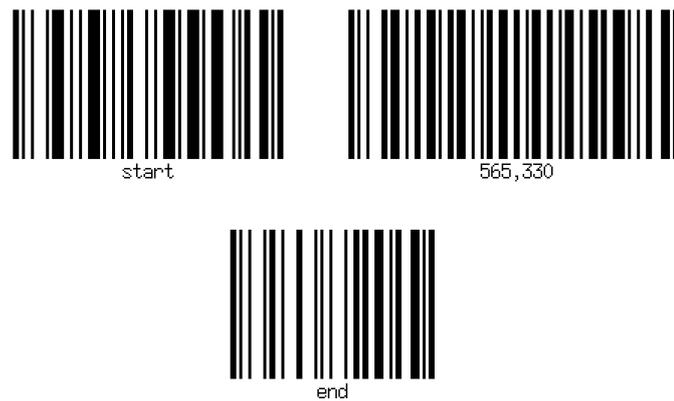


Figura 3.4: Códigos de barras da mensagem embutidos nos quadros de vídeo transmitidos pela tela. O segundo código de barras contém as dimensões físicas da tela ( $L_x, L_z$ ) em milímetros, separadas por uma vírgula.

## 3.2 Calibração da câmera

Para a calibração da câmera, cuja função é reduzir efeitos de distorção oriundos dos parâmetros intrínsecos do sensor óptico, foi implementado um código de calibração responsável por calcular uma matriz de transformação otimizada que possa ser aplicada aos quadros capturados para determinada câmera e remover o máximo possível das distorções provocadas pelos parâmetros intrínsecos, conforme o método realizado pela OpenCV para tal calibração. (OPENCV CONTRIBUTORS, n.d.)

Assumindo um modelo ideal de câmera *pinhole* com grades de pixels retangulares, a matriz de calibração  $H$  (aprofundada na Seção 2.9) é dada por:

$$\mathbf{H} = \begin{bmatrix} f/l_x & 0 & u_0 \\ 0 & f/l_z & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (3.1)$$

onde  $f$  é a distância focal da câmera,  $(u_0, v_0)$  são as coordenadas da interseção dos eixos ópticos com o plano da imagem (ponto principal), e  $(l_x, l_z)$  são os tamanhos verticais e horizontais dos pixels do IS, respectivamente (ZHANG, Z., 2000; MARCHAND; UCHIYAMA; SPINDLER, 2016).

A estimativa de  $\mathbf{H}$  é posteriormente necessária para corrigir a distorção de cada quadro do vídeo, além de ser usada como parâmetro para a estimativa da pose da câmera e das coordenadas relativas à tela através da função *solvePnP* do OpenCV.

Além de  $\mathbf{H}$ , o processo de calibração também retorna os coeficientes de distorção que devem ser usados em conjunto com tal matriz. Esses coeficientes representam a distorção óptica introduzida pelas lentes da câmera. Eles são usados para corrigir imagens capturadas e transformá-las para uma perspectiva idealizada, onde linhas retas no mundo real aparecem como retas na imagem.

As lentes de uma câmera introduzem dois tipos principais de distorção (OPENCV CONTRIBUTORS, n.d.):

- Distorção Radial: As linhas retas parecem curvas, especialmente nas bordas da imagem (efeito "olho de peixe").
- Distorção Tangencial: Ocorre devido a desalinhamentos na montagem da lente, fazendo com que os pixels se desloquem de forma não radial.

Para nossa aplicação com a calibração via OpenCV, esses coeficientes são armazenados conforme a matriz  $D$  3.2, sendo  $k_i$  os coeficientes da distorção radial, e  $p_i$  os coeficientes de distorção tangencial:

$$\mathbf{D} = \begin{bmatrix} k_1 & k_2 & p_1 & p_2 & p_3 \end{bmatrix}. \quad (3.2)$$

Tomando o fluxograma da Figura 3.5, código funciona da seguinte maneira: uma captura de vídeo feita pela câmera que deseja-se calibrar é usada como entrada e pode ser feita previamente ou no momento de execução do código (bloco 1). Para a segunda opção, a câmera é aberta como uma entrada de *webcam* no código e portanto sua conexão deve ser adaptada para funcionar de tal maneira. Por exemplo, para capturar ao vivo com um *smartphone*, foi utilizado o software Droidcam disponível na Playstore.

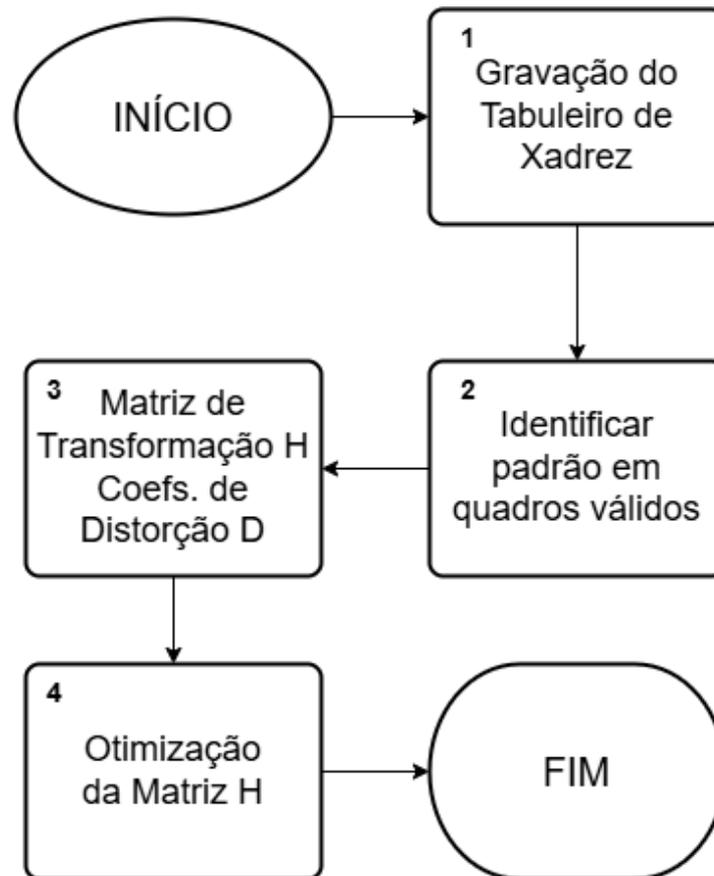


Figura 3.5: Fluxograma do código de calibração da câmera para extração dos parâmetros intrínsecos

Esse vídeo capturado possui como foco um tabuleiro de xadrez qualquer (*chessboard pattern*). As dimensões do tabuleiro são irrelevantes, desde que todo o quadriculado seja quadrado. Ao longo do vídeo de calibração, deve-se variar de forma arbitrária a posição e orientação desse tabuleiro para capturar a maior quantidade possível de maneiras que ele pode ser observado pela câmera em um ponto fixo. É recomendado realizar a variação de forma lenta e com pequenas pausas de alguns segundos (geralmente 2 a 3 segundos) a cada nova orientação, para aumentar as chances do algoritmo de detecção do tabuleiro identificá-lo em pelo menos alguns quadros.

Com o vídeo carregado em memória, o código processa quadro a quadro a identificação do tabuleiro e desenha com linhas coloridas sua orientação (bloco 2 Figura 3.9). Essa identificação toma como base as quinas de cada quadrado dentro do tabuleiro para comparar as dimensões identificadas com as estipuladas como válidas. Se uma grade é identificada e possui as dimensões  $m \times n$  esperadas, considera-se que é o tabuleiro em questão, como visto na Figura 3.6.

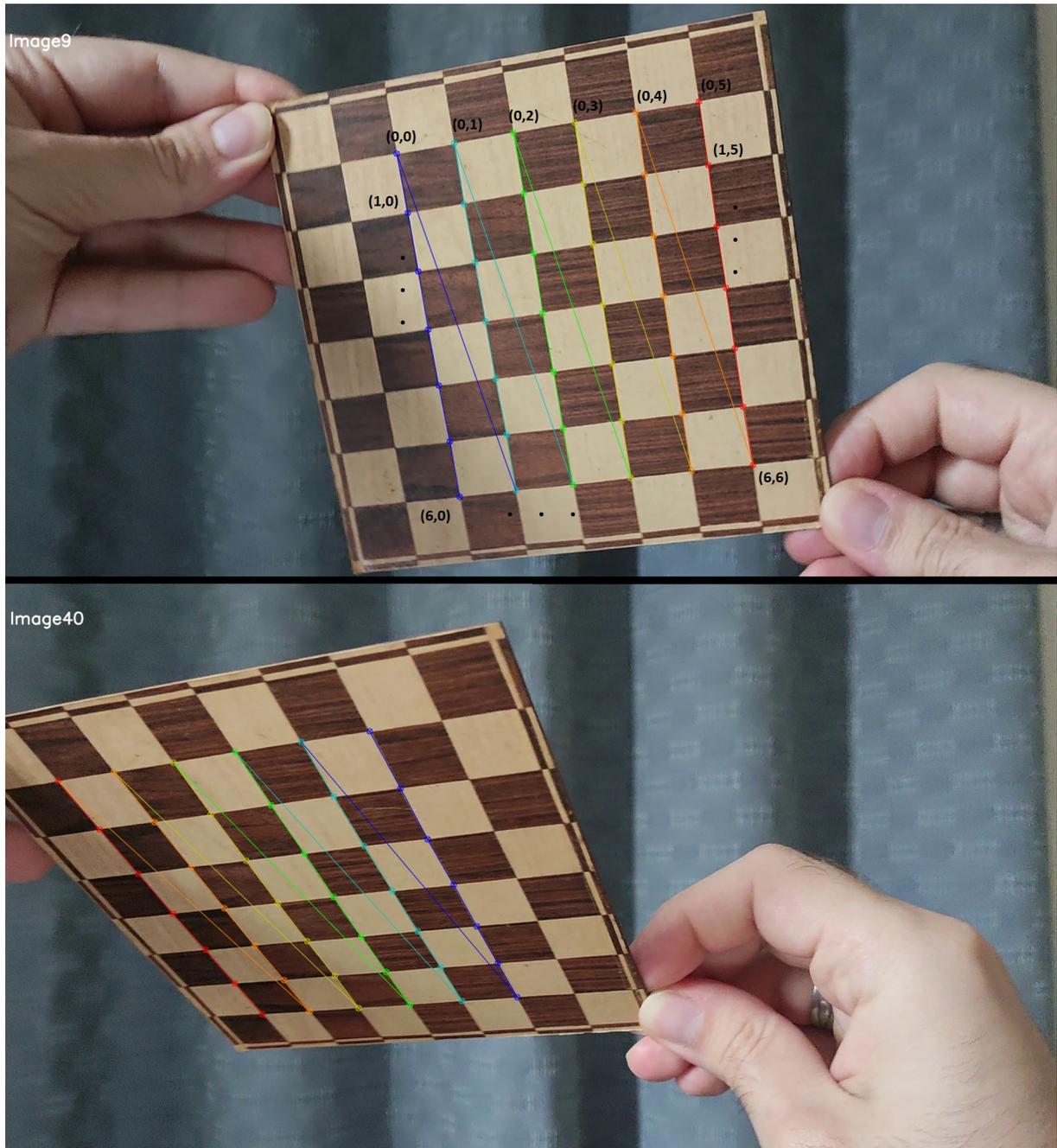


Figura 3.6: Dois quadros exemplificando a identificação do tabuleiro de xadrez com uma matriz  $6 \times 7$ . No primeiro quadro, alguns pontos estão sinalizados para exemplificar a atribuição dos pontos de objeto.

São armazenados os quadros onde o tabuleiro foi identificado corretamente em um novo *array* que é usado para alimentar a função que calcula a matriz de homografia  $H$  da câmera (bloco 3). Essa função recebe como parâmetros os pontos de todo o *grid* identificado na imagem, chamados de *image points*; Os pontos do objeto são dados da seguinte forma: assumindo-se que uma extremidade do tabuleiro identificado seja sua coordenada de origem no espaço, são informadas as posições dos demais vértices de seu quadriculado relativas a

essa origem. Como as dimensões verdadeiras do tabuleiro são desconhecidas e foi assumido que ele é um quadrado perfeito, esses pontos de objeto (*object points*) são passados como incrementos unitários em ambas as dimensões vertical e horizontal.

Com ambas as informações dos *image points* e *object points*, a função realiza o cálculo da matriz de transformação para correção da câmera, estruturada conforme a Equação 2.13, cuja forma simplificada para esta aplicação é dada conforme a Eq. 3.1.

Ao associar as coordenadas e orientação dos *image points* extraídos pela identificação do tabuleiro com as posições relativas unitárias dos *object points*, é possível identificar o quão divergente determinado ponto está de sua posição ideal, caso a lente da câmera não apresentasse efeitos de distorção e oclusão. A partir dessa amostragem, são calculados numericamente os valores da matriz  $H$  e os coeficientes de distorção  $D$  que melhor se encaixem para corrigir todo o quadro capturado. Em seguida, foi realizada uma etapa de otimização da matriz para pequenos ajustes de arredondamento (bloco 4).

Daqui em diante, toda captura utilizada no restante do código passará por uma função de correção que recebe a imagem do quadro, a matriz otimizada e os coeficientes de distorção para correção da distorção quadro a quadro.

Para validação visual ao aplicar a correção de distorção, exibe-se um quadro do vídeo capturado após a correção na Figura 3.7. (OPENCV CONTRIBUTORS, n.d.)



Figura 3.7: Resultado pós calibração da câmera para correção das distorções oriundas dos parâmetros intrínsecos.

### 3.3 Pré processamento dos quadros

O algoritmo para realizar um pré tratamento dos quadros do vídeo capturado a fim de identificar uma tela que esteja transmitindo uma sequência de código de barras com uma mensagem codificada é composto pelas etapas da Figura 3.8 da seguinte forma:

- 1: leitura do quadro atual a ser processado, armazenando-o em memória como um *Pandas pd.array* (ORG, 2024);
- 2: correção das distorções provenientes da câmera, utilizando a matriz de transformação obtida durante a calibração descrita na Seção 3.2;
- 3: conversão do padrão de cores para escala de cinza e binarização, isolando os elementos claros/luminosos da imagem, empregando o método *threshold* de Otsu do OpenCV (OPENCV, 2024c);
- 4: engrossamento de contornos escuros para otimizar a continuidade de formas geométricas, utilizando os métodos de dilatação e erosão do OpenCV (OPENCV, 2024a).

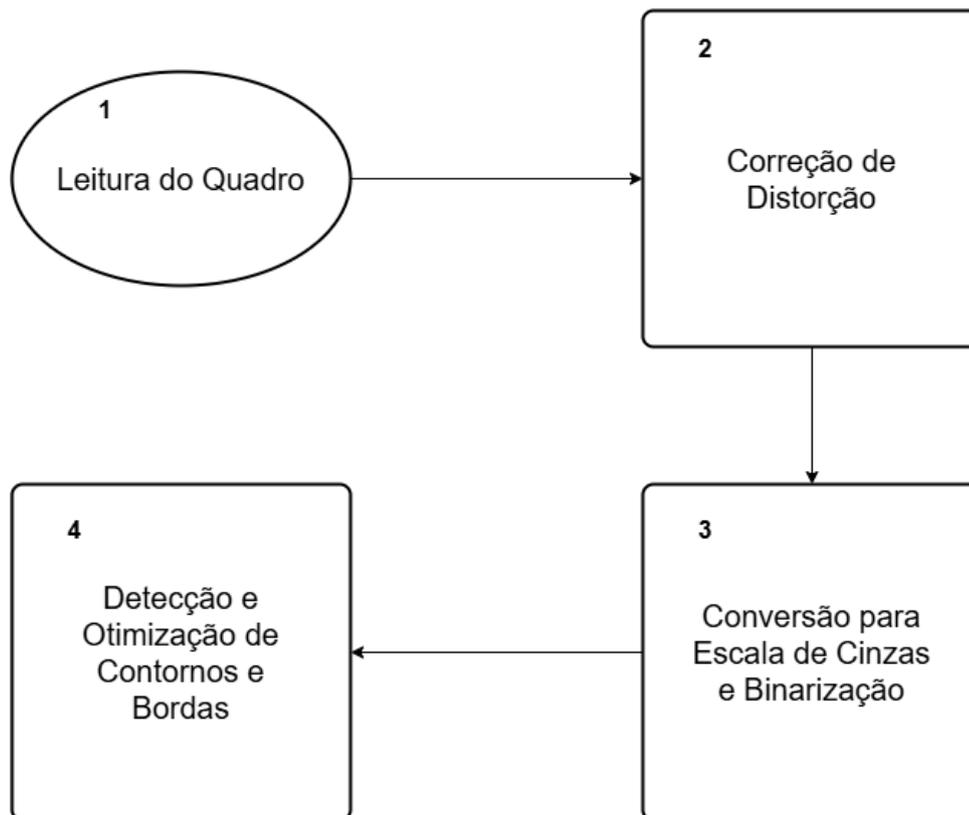


Figura 3.8: Fluxograma da etapa de pré processamento dos quadros.

Para aumentar a assertividade da identificação da tela e do código de barras, foi realizado um processo de binarização da imagem já em escala de cinza. Tal processo consiste em escolher um valor de limiar (*threshold*) entre 0 e 255 para 8 bits de informação, e pixel a pixel, alterar sua intensidade completamente para 0 ou para 255 caso seu valor original esteja acima ou abaixo do *threshold*.

Realizar a binarização permite remover parcialmente a interferência de pontos vizinhos aos que deseja-se identificar, pois elimina efeitos de sombra ou dispersão de luz que geram pixels com diferentes intensidades que podem dificultar tais identificações.

Para melhor destaque dos contornos e posteriormente dos códigos de barras, foram utilizados elementos estruturantes em forma de retângulos de  $3 \times 3$  pixels para aplicar uma dilatação morfológica conforme a Equação 2.3. Dessa forma, efeitos de distorção e serrilhados interferem menos na identificação das barras. Por outro lado, uma desvantagem é a perda de distância máxima que a câmera pode estar desse código de barras, pois com contornos mais grossos, dimensões menores provocam sobreposição entre esses traços, descaracterizando a forma do código.

### 3.4 Identificação de regiões de interesse

O processo apresentado na Figura 3.9 inicia com a aplicação de um código de detecção de bordas (*edge detection*) de Canny, conforme aprofundado na Seção 2.8. É preciso analisar se essas identificações estão dentro de uma margem aceitável para o restante do código, ou se são necessários ajustes dos coeficientes de *threshold* superior e inferior aplicados no algoritmo de Canny (bloco 1 Figura 3.9).

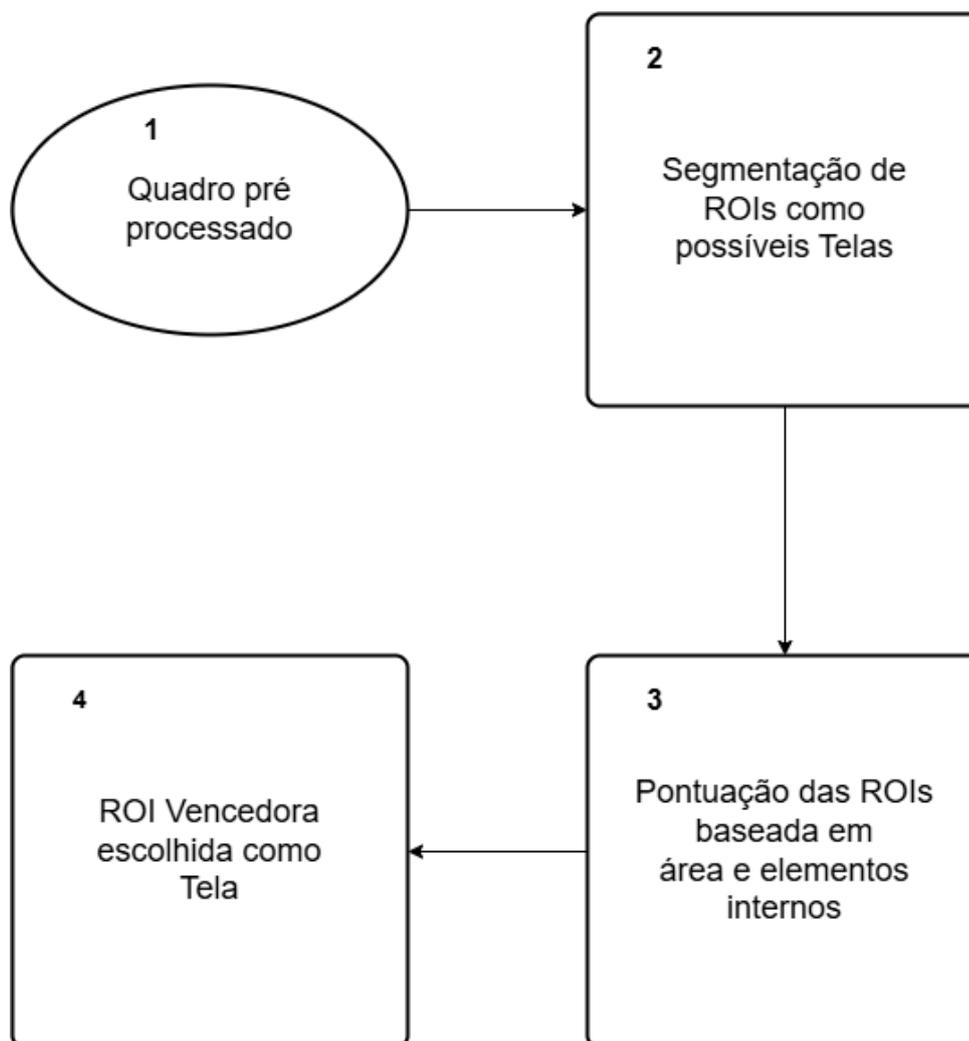


Figura 3.9: Fluxograma com as etapas da determinação de regiões de interesse e escolha de qual delas é a tela.

Muitas vezes os resultados para determinada região da imagem se apresentam de forma segmentada. No caso da identificação da tela, na maioria dos testes, entre 80 e 90% das bordas foram identificadas corretamente, porém com diversas interrupções entre seus contornos e tracejados irregulares, curvados ou com desnível, ao invés de linhas retas. Para formar uma borda completamente conectada e com traços retos, foi aplicada uma dilatação morfológica

nesses traços e um código de identificação de contornos (*findContours*) do OpenCV (OPENCV, 2024b), o qual é baseado no **Algoritmo de Suzuki e Abe** (SUZUKI et al., 1985). Tal método realiza a análise topológica da imagem ao percorrer bordas e buracos presentes na imagem de forma sistemática, utilizando-se de um conceito de bordas externas e internas. Este método evita a necessidade de analisar todos os pixels da imagem, focando apenas nas bordas, o que o torna altamente eficiente para aplicações em visão computacional e processamento de imagens.

O resultado de um quadro pós processamentos é exibido na Figura 3.10

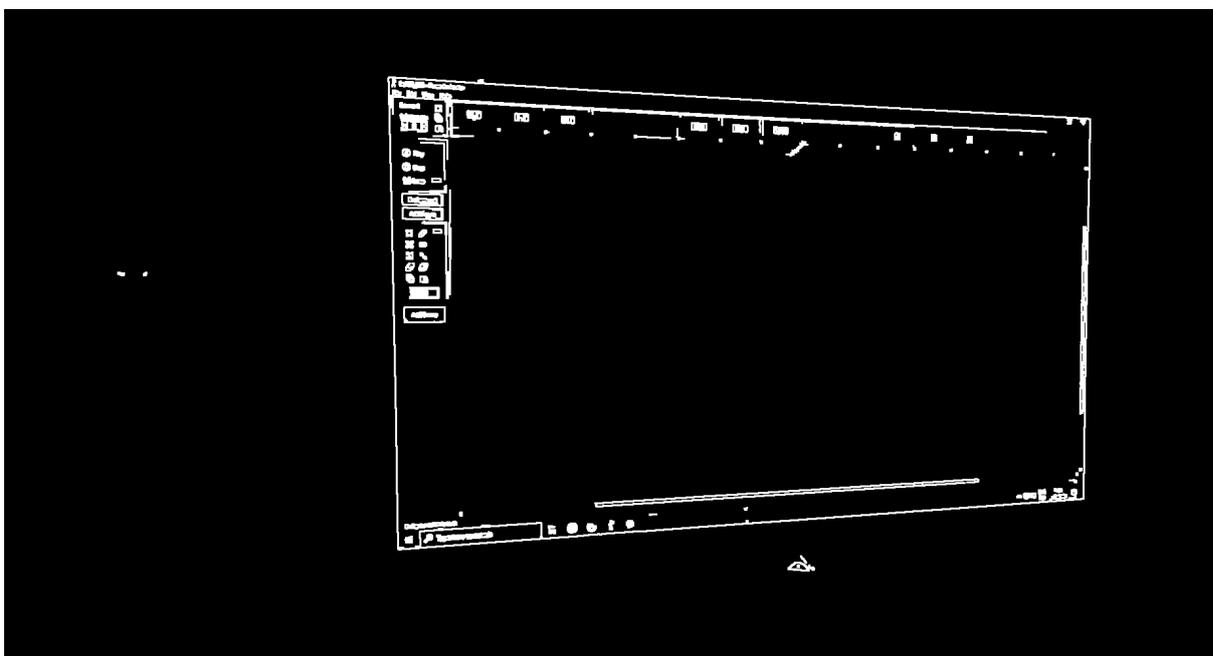


Figura 3.10: Quadro binarizado com dilatação morfológica após identificação dos contornos

Com as novas regiões delimitadas de forma mais refinada, foi aplicada uma demarcação com caixas delimitadoras ao redor de cada Região de Interesse - *Region of Interest* (ROI), para obter retângulos com dimensões equivalentes aos pontos mais externos de cada contorno identificado (segmentação das ROIs). Por exemplo: se uma tela e seu contorno estão em uma posição diagonal e inclinada, a caixa delimitadora (*bounding box*) será um retângulo com dimensões suficientes para beirar as quinas mais distantes do contorno. Sendo assim, cada *bounding box* obtida com os contornos da imagem é armazenada como uma região a ser processada para identificar qual contém a tela, como na Figura 3.11 (bloco 2).

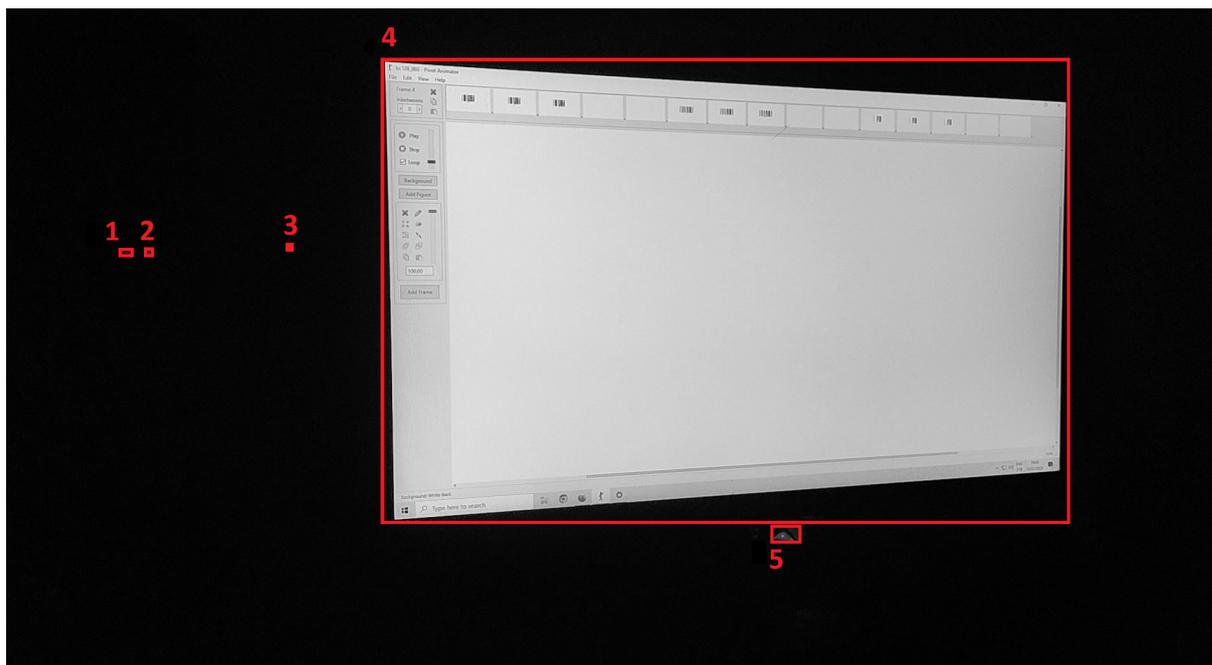


Figura 3.11: *Bounding boxes* traçadas em vermelho para os contornos identificados como possíveis telas no quadro.

### 3.5 Identificação da tela

Uma rotina no terceiro bloco da Figura 3.9 percorre região a região para analisar se ela contém uma possível tela. A primeira etapa da análise consiste em determinar a quantidade de pixels com intensidade superior a um valor estabelecido, e calcular o percentual dessa região composto por esses pixels. Se esse percentual for igual ou superior ao configurado pelo usuário do sistema, tal região prossegue para a segunda etapa da análise. Esse percentual foi determinado de maneira empírica e deve ser ajustado para condições de gravação diferentes das estipuladas para este projeto.

Com todas as regiões que possivelmente contém uma tela, a próxima etapa do algoritmo consiste em atribuir uma pontuação para cada região, baseado em quantas outras regiões estão contidas em cada uma. Se uma região não contém nenhuma outra, sua pontuação é 1. Para cada outra que esteja totalmente dentro da região em análise, é somado mais 1 ponto. Ao final do processamento (bloco 4 Figura 3.9), a que tiver a maior pontuação é considerada a com maior probabilidade de ser a tela. Em caso de empate, a que estiver salva no menor índice do *array* de regiões será considerada como a correta.

Tal lógica parte do suposto que dentro da tela haverá uma maior quantidade de *bounding boxes* do que em outros objetos luminosos, pois há mais elementos que também geraram regiões, como os códigos de barras transmitidos e outras informações visuais contidas no

monitor. Nas luzes de caixas de som de computador, por exemplo, além de suas dimensões serem menores, não há outras informações contidas dentro da luz dos LEDs.

Na gravação demonstrada na Figura 3.11, a BBOX 4 é escolhida corretamente como a tela. É importante enfatizar que esta abordagem foi utilizada apenas na identificação de telas deste desenvolvimento. O método empregado para a identificação da tela no Capítulo 5 é outro, e será devidamente aprofundado dentro de tal Capítulo.

### 3.6 Decodificação das mensagens para estimação da distância

Apenas a informação de onde está o monitor na imagem não é suficiente para o sistema de posicionamento. São necessários dados das dimensões da tela para calcular a distância que a câmera está dessa tela e suas coordenadas relativas de acordo com a orientação da captura. Para a distância, a informação das dimensões físicas do objeto e uma constante de correção são suficientes para aplicar em um modelo que relaciona tais dimensões físicas com as dimensões virtuais que a *bounding box* do objeto ocupa na tela. Essa constante representa as distorções provocadas pelos parâmetros intrínsecos da câmera, sendo que mesmo após a correção de distorção, ainda é necessário mais este refinamento para reduzir os erros.

A aproximação das dimensões virtuais baseada na *bounding box* propaga erros como variáveis dependentes da inclinação e orientação da tela, pois como a região delimitada pelo retângulo inclui a maior distância entre as extremidades, dependendo do ângulo da tela, maior a distorção de perspectiva e discrepância entre as dimensões da BBOX e as da tela.

Para processar a mensagem transmitida, foi aplicado um código pronto de identificação e leitura para esse tipo de código BC-128 com a biblioteca Pyzbar, dentro da *bounding box* selecionada como contendo o monitor. Devido a informação estar armazenada de forma temporal e ser retransmitida ciclicamente, uma pequena parte da gravação é separada em memória (*frames buffer*), de tamanho suficiente para conter a mensagem completa. Por exemplo: se a mensagem é completamente transmitida em 100 quadros, o *buffer* deve conter ao menos 100 *slots* para 100 quadros. Entretanto, não é possível saber em que instante da mensagem a captura desses quadros iniciou, portanto a quantidade armazenada deve ser ao menos o dobro do tamanho da mensagem, para caso a captura tenha iniciado no final de um bloco de mensagens.

Em seguida, foi então aplicado um algoritmo que remove quadros de redundância usados para compensar instabilidades de captura na câmera, para ter como resultado apenas sequências estruturadas da seguinte forma:  $[..L_x, L_z \text{ "END" "START" } L_x, L_z \text{ "END" } ..]$ , sendo  $L_x$  e  $L_z$  as dimensões horizontais e verticais em mm da tela, respectivamente. Esse *array* é então percorrido até encontrar uma ocorrência do 'START', as mensagens das dimensões x e y em milímetros, e finalizar com 'END' na sequência correta, para considerar como uma mensagem completa. Na Figura 3.12 é possível visualizar um quadro analisado após as etapas da Seção 3.3 com um código de barras informando o fim de uma sequência de mensagens ('END'). Apesar do código de barras exibido estar incompleto e com pequenas sobreposições mesmo após o pré processamento, o sistema já foi capaz de trabalhar com o quadro desta forma.

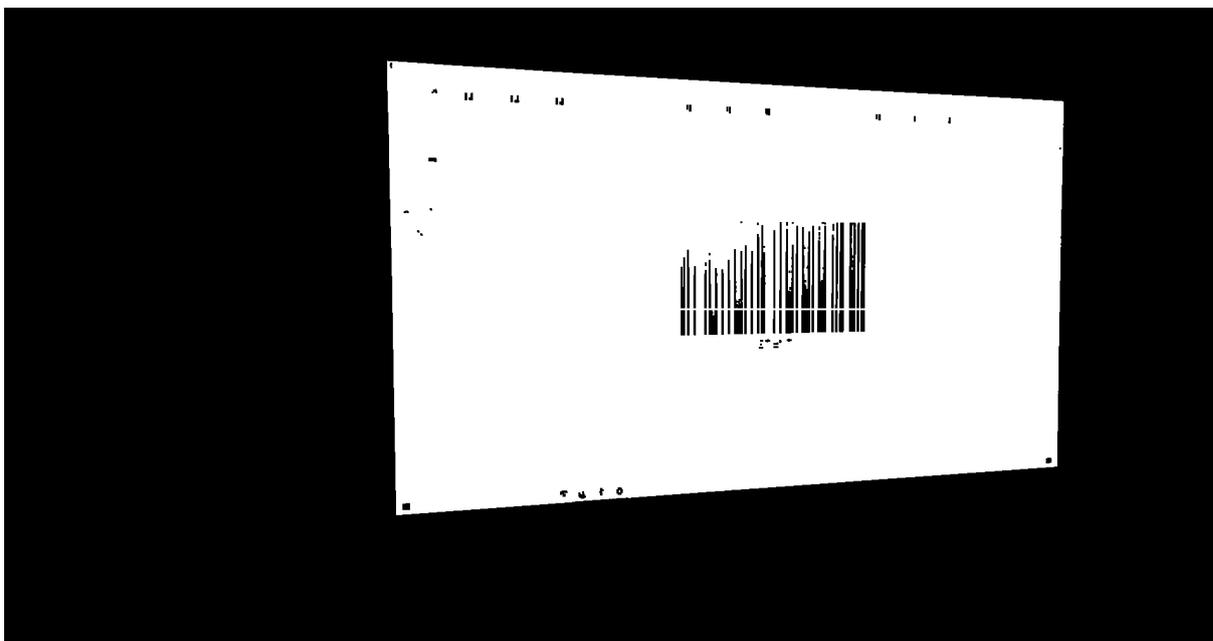


Figura 3.12: Quadro binarizado com erosão morfológica para ênfase no código de barras.

### 3.7 Obtenção da distância, pontos virtuais e reais

Com o processamento da mensagem dos quadros e a extração das dimensões do *display*, é possível calcular a distância  $d$  da câmera em relação ao centro da tela definida pela *bounding box*, agora conhecida com as dimensões delas. O cálculo aplicado é uma relação entre a altura física  $H_s$  da ROI, a dimensão vertical  $H_f$  da imagem em pixels, a altura virtual  $H_b$  da ROI em pixels e uma constante de correção  $C$ .

É possível estabelecer o cálculo da distância  $d$  conforme a Equação 3.3.

$$d = \frac{C \cdot (H_s \cdot H_f)}{H_b} \quad (3.3)$$

A constante de correção  $C$  depende do tamanho do sensor e da distância focal da lente. Na Figura 3.13, a visão real é a realidade com a câmera e a tela posicionados no espaço físico, enquanto que a visão virtual é um mapeamento do que seria a posição da câmera nas coordenadas virtuais da imagem capturada. Há uma representação da câmera a uma distância virtual  $D$  da tela centralizada em seu quadro (visão virtual), e outra representação para uma distância real  $d$  (visão real), sendo ambas as distâncias desconhecidas a priori. Note que está representada somente a metade superior do campo de visão da câmera, portanto há a altura real  $H_s/2$  da ROI, a altura vertical virtual  $H_f/2$  da resolução da câmera, e a altura virtual  $H_b/2$  da tela/ROI.

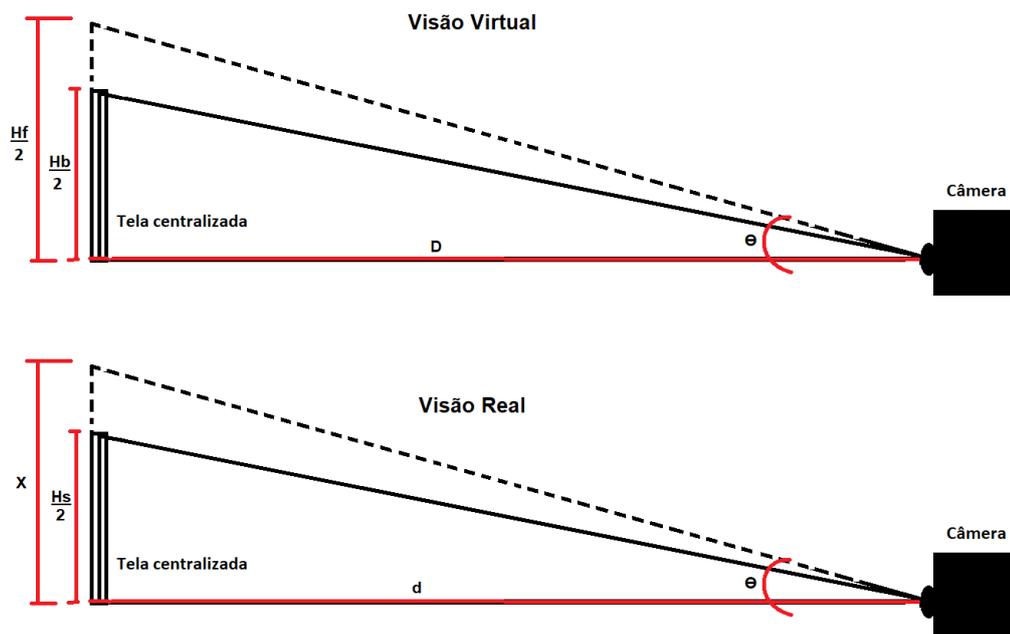


Figura 3.13: Relações de distâncias e dimensões entre a câmera e a tela.

A partir da Figura 3.13, é possível estabelecer uma relação por semelhança de triângulos entre as alturas  $X$  e  $H_f/2$  da seguinte forma:

$$\frac{X}{\left(\frac{H_s}{2}\right)} = \frac{\left(\frac{H_f}{2}\right)}{\left(\frac{H_b}{2}\right)} \quad (3.4)$$

É possível simplificar as frações e isolar  $X$  conforme a Equação 3.5:

$$X = \frac{H_f \cdot H_s}{2H_b} \quad (3.5)$$

Tomando a Equação 3.3 e substituindo a razão de alturas por  $X$  da Equação 3.5, obtém-se:

$$d = \frac{C \cdot (H_s \cdot H_f)}{H_b} = C \cdot 2X \quad (3.6)$$

Isolando  $C$ :

$$C = \frac{d}{2X} \quad (3.7)$$

Como a Equação 3.7 representa a divisão do cateto adjacente  $d$  pelo dobro do cateto oposto  $X$ , conforme a Figura 3.13,  $C$  pode ser obtido da seguinte forma:

$$C = \frac{1}{2 \tan(\theta)} \quad (3.8)$$

Portanto,  $C$  é dado em função do ângulo de visão  $\Psi$  da câmera, sendo  $\theta = \Psi/2$ .

O parâmetro  $C$  é diferente para cada câmera e deve ser obtido empiricamente caso  $\Psi$  seja desconhecido. A calibração consiste em capturar uma imagem da tela transmissora posicionada a uma distância conhecida  $d$ , garantindo um enquadramento nivelado e centralizado.

Se  $H_f$ ,  $H_s$  e  $H_b$  não mudam, para cada  $C$  é obtido um  $d$  equivalente. Sendo assim, ocorre a relação dada pela Equação 3.9:

$$\frac{C_0}{d_0} = \frac{C_1}{d_1} \quad (3.9)$$

Assumindo  $C_1 = 1$  na Equação (3.9) e isolando  $C_0$ , obtém-se:

$$C_0 = \frac{d_0}{d_1} \quad (3.10)$$

Portanto, ao executar o código de posicionamento com a Equação 3.3 e  $C = C_1 = 1$ , é gerada uma estimativa de distância incorreta  $d_1$ . O fator de correção é então dado pela razão entre a distância real e a calculada, conforme a Equação (3.10), permitindo determinar o valor correto de  $C = C_0$ . Ao substituir esse valor  $C_0$  recalibrado em (3.3), a distância estimada passa a refletir corretamente a realidade, onde  $d_0 = d$ .

Além da informação sobre a distância, há mais duas informações extraídas pelo código abordado nesta Seção: os pontos virtuais da tela na imagem e os pontos reais da tela em milímetros. Para os pontos virtuais (*Image Points*), foi desenvolvida uma função que detecta as quatro quinas do contorno identificado da tela, calcula os pontos intermediários entre cada aresta, e armazena suas coordenadas virtuais em uma matriz  $4 \times 2$  (4 vértices com as coordenadas  $x$  e  $z$ ). Já para os pontos reais (*object points*), com base nas dimensões informadas de largura ( $L_x$ ) e altura ( $L_z$ ) da tela, foi armazenado em outra matriz  $4 \times 2$  na mesma ordem da matriz de *Image Points* para manter a correta relação entre eles nas etapas adiante, de tal forma que os pontos de objeto (*Object Points*) OP seguem esta ordem, seguindo as coordenadas da Figura 3.1:

$$\begin{aligned}
 OP = \{ & (L_x/2, L_z/2); (0, L_z/2); (-L_x/2, L_z/2); \\
 & (L_x/2, 0); (-L_x/2, 0); \\
 & (L_x/2, -L_z/2); (0, -L_z/2); (-L_x/2, -L_z/2) \}. \quad (3.11)
 \end{aligned}$$

### 3.8 Cálculo das coordenadas

O passo final (Bloco 7) na Figura 3.2 envolve calcular as coordenadas da câmera em relação à tela. O objetivo dessa rotina é recuperar as coordenadas da câmera em um espaço tridimensional utilizando os eixos  $X$ ,  $Y$  e  $Z$ , com o centro da tela servindo como ponto de origem. O eixo  $X$  representa o deslocamento horizontal,  $Y$  representa a profundidade e  $Z$  representa o deslocamento vertical, conforme mostrado na Figura 3.1.

Para determinar a pose da câmera com o método PnP apresentado na Seção 2.11, foi utilizada a função *solvePnP* do OpenCV (TEAM, 2024). Tomada a matriz de calibração  $H$  da câmera, com a qual a projeção em perspectiva pode ser dada por:

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \mathbf{H}\mathbf{\Pi}\mathbf{T} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}, \quad (3.12)$$

onde  $\mathbf{\Pi}$  é uma matriz de projeção dada por: (MARCHAND; UCHIYAMA; SPINDLER, 2016)

$$\mathbf{\Pi} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad (3.13)$$

e  $\mathbf{T}$  é composta por um vetor de rotação  $\mathbf{r}$  e um vetor de translação  $\mathbf{t}$  como:

$$\mathbf{T} = \begin{pmatrix} \mathbf{r} & \mathbf{t} \\ 0 & 1 \end{pmatrix}. \quad (3.14)$$

Os quatro pares de pontos dos vértices detectados ao final da Seção 3.7, ou seja,  $i = 1, 2, 3, 4$  da tela permitem resolução do problema PnP com  $n = 4$  pontos, resultando em uma solução única para a posição estimada. O *Perspective-3-Point* (P3P) é um caso específico do PnP aprofundado na Seção 2.11 que resolve o problema apenas com 3 pontos conhecidos. Entretanto, geometricamente, com apenas 3 pontos, o problema da pose da câmera não tem solução única, podendo haver até 4 soluções possíveis devido a ambiguidades geométricas na perspectiva. Ao informar um quarto ponto de referência (*Perspective-4-Point* (P4P)), é introduzida uma restrição adicional que elimina as ambiguidades, permitindo determinar uma única solução consistente (MARCHAND; UCHIYAMA; SPINDLER, 2016). É possível observar a correspondência dos quatro vértices da tela real em coordenadas virtuais na Figura 3.14.

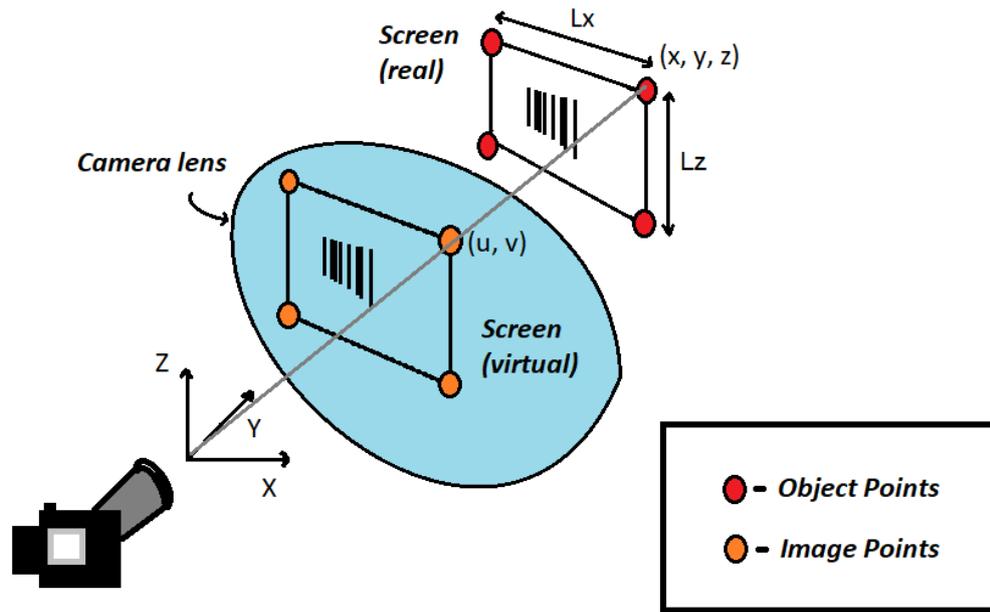


Figura 3.14: Problema P4P para estimativa de pose, com quatro pontos de referência

A função *solvePnP* do OpenCV, baseada no trabalho de (GAO et al., 2003), funciona com os quatro pares de pontos e retorna tanto  $\mathbf{r}$  quanto  $\mathbf{t}$  da Eq. (3.14). Posteriormente,  $\mathbf{r}$  é transformado em uma matriz de rotação  $\mathbf{R}$  utilizando a fórmula de rotação de Rodrigues (DAI, 2015), que será usada mais adiante para calcular as coordenadas da câmera.

A fórmula de Rodrigues aplicada pelo OpenCV é usada para converter um vetor de rotação (também chamado de vetor de eixos e ângulo) em uma matriz de rotação (GALLEGO; YEZZI, 2014). É muito usada em gráficos 3D, robótica, simulações físicas e algoritmos de rotação sem a necessidade de lidar com quaternions.

A matriz de rotação  $\mathbf{R}$ , baseada na Fórmula de Rodrigues, para um ângulo  $\theta$  em torno de um eixo unitário  $\mathbf{r} = (r_x, r_y, r_z)$  é dada por:

$$\mathbf{R} = \begin{bmatrix} \cos \theta + r_x^2(1 - \cos \theta) & r_x r_y(1 - \cos \theta) - r_z \sin \theta & r_x r_z(1 - \cos \theta) + r_y \sin \theta \\ r_y r_x(1 - \cos \theta) + r_z \sin \theta & \cos \theta + r_y^2(1 - \cos \theta) & r_y r_z(1 - \cos \theta) - r_x \sin \theta \\ r_z r_x(1 - \cos \theta) - r_y \sin \theta & r_z r_y(1 - \cos \theta) + r_x \sin \theta & \cos \theta + r_z^2(1 - \cos \theta) \end{bmatrix} \quad (3.15)$$

onde:

- $\mathbf{r} = (r_x, r_y, r_z)$  é o vetor unitário de rotação.

- $r_x, r_y, r_z$  são as componentes desse eixo nos eixos  $x, y, z$ , respectivamente.
- Esse vetor deve ser unitário, ou seja:

$$\|\mathbf{k}\| = \sqrt{r_x^2 + r_y^2 + r_z^2} = 1$$

- $\theta$  é o ângulo de rotação em radianos.
  - $\theta > 0$  significa rotação no sentido anti-horário em torno do eixo  $\mathbf{r}$ .
  - $\theta < 0$  significa rotação no sentido horário.

Sendo assim, dada a transformação de  $\mathbf{r}$ ,  $\mathbf{R}$  é então usada para obter a estimativa da posição real  $(x_c, y_c, z_c)$  da câmera nas coordenadas do mundo (*World Coordinates*) através da rotação no sentido inverso da coordenada central da tela:

$$\begin{bmatrix} x_c & y_c & z_c \end{bmatrix} = \mathbf{R}^{-1}\mathbf{t}. \quad (3.16)$$

Ao aplicar a inversa de  $\mathbf{R}$  sobre  $\mathbf{t}$ , o sistema de coordenadas local da câmera  $\mathbf{t}$  (onde o objeto está na origem) foi transformado de volta para o sistema de coordenadas do mundo, permitindo a estimativa da posição da câmera no mundo.

## Capítulo 4

### Resultados - S2CP com códigos de barras

Nesta Seção são apresentados os resultados do sistema S2CP apresentado no Capítulo 3, em diferentes posicionamentos (ângulos e distâncias) da câmera. O objetivo é avaliar a precisão do sistema na estimativa da posição da câmera em relação à tela no espaço 3D.

Para essa validação, foi conduzido um experimento em condições reais usando um monitor de computador pessoal (PC) Ozone®OZDSP25PRO de tela plana 56.5 cm × 33.0 cm, com resolução 1920 × 1080 px e 144 Hz de taxa de atualização como a tela imóvel. Os vídeos transmitidos são do formato *Audio Video Interleave* (AVI), a 30 FPS. Foi utilizado um smartphone POCO®F5 com câmera principal de 64 MP, ajustada para capturar em 1920 x 1080 pixels @ 30 FPS, como receptor. Os códigos de barras transmitidos possuem dimensões de 18,0 cm x 9,0 cm, ocupando aproximadamente 9% da área da tela. Utilizar um FPS baixo, combinado com uma boa identificação visual dos códigos de barras amenizam significativamente possíveis perdas de sincronia perceptivas.

A Equação (4.1) foi posteriormente usada na Equação (3.12) para resolver o sistema  $P4P$ . A distância entre a tela e a câmera varia de 0 a 125 centímetros ao longo do eixo  $Y$ , com posicionamento vertical câmera (eixo  $Z$ ) variando até 20 cm, e o posicionamento horizontal (eixo  $X$ ) em até 50 cm.

Para a calibração da câmera através do método apresentado na Seção 3.2, foi encontrada a matriz  $H$ , dada por 3.1, igual a

$$\mathbf{H} = \begin{bmatrix} 1.7852 & 0.0000 & 0.9707 \\ 0.0000 & 1.7903 & 0.5508 \\ 0.0000 & 0.0000 & 1.0000 \end{bmatrix}. \quad (4.1)$$

Os coeficientes de distorção radial e tangencial (conforme aprofundado na Seção 3.2) foram:

$$\mathbf{D} = \begin{bmatrix} -0.05314232 & 0.20946573 & 0.00310676 & 0.00614488 & -0.22711199 \end{bmatrix}. \quad (4.2)$$

As gravações dos vídeos foram realizadas visando garantir o alinhamento e posicionamento desejado entre a câmera e a tela. Para tal, foi utilizado um tripé ajustável como suporte para o *smartphone*. Apesar desses ajustes manuais, ainda ocorreram pequenos desalinhamentos devidos a estrutura irregular do tripé que não apresenta ajustes para a angulação do *smartphone* fixo. Na Figura 4.1 é possível visualizar a estrutura e disposição dos materiais utilizados para o experimento.



Figura 4.1: Demonstração do ambiente de gravação. O *smartphone* está fixado a um tripé ajustável para poder ser posicionado adequadamente em relação a tela transmissora dos códigos de barras.

Sendo assim, foi desenvolvida uma interface própria de edição de vídeos que possibilita corrigir manualmente esses pequenos desalinhamentos. O sistema desenvolvido utiliza uma interface gráfica interativa para aplicar transformações em um quadro do vídeo como referência, permitindo a edição e ajuste da perspectiva do vídeo através da matriz de homografia. Tal *software* encontra-se no Apêndice (A).

Todas as gravações realizadas foram dentro de um cômodo sem iluminação, exceto pela luz da própria tela e de pequenos LEDs de equipamentos eletrônicos aos arredores, como caixas de som, teclado e mouse. Todas as gravações contêm apenas um monitor, e a imagem transmitida pelo monitor é um quadro branco com a sequência de código de barras sendo transmitida de forma temporal, centralizada no monitor.

## 4.1 Resultados usando sistema P4P

De 20 gravações realizadas, em 17 o algoritmo identificou corretamente a tela e mensagem, permitindo nestas o correto processamento da mensagem e cálculo do posicionamento. Para visualização do posicionamento calculado, ao final do processo é plotado um gráfico 3D com os eixos e origem descritas partindo do centro do monitor, e um ponto representando a posição estimada da câmera utilizando o método da Seção 3.8. Para as tabelas desta seção, as células em azul possuem valor de diferença absoluta igual ou inferior a 5 centímetros, considerado um erro muito baixo para este protótipo, tomando como comparação o erro médio de 6,5 cm apresentado por (FENG et al., 2023).

Para calcular o erro médio absoluto  $MAE$  com as diferenças absolutas dos três eixos, utiliza-se a Equação 4.3 com as diferenças absolutas totais e  $N = 17$  gravações:

$$MAE^{(A)} = \frac{\sum_{i=1}^{17} e_i^{(A)}}{17}. \quad (4.3)$$

As Tabelas 4.1, 4.2 e 4.3 trazem respectivamente as coordenadas reais  $A = \{X, Y, Z\}$  para todas as 17 gravações, as estimativas calculadas  $\hat{A}_i$ , e o erro absoluto  $e_i^{(A)}$  entre os valores corretos e calculados.

Para a Tabela 4.1, a orientação utilizada no posicionamento horizontal da câmera em relação à tela tomando os eixos de coordenadas da Figura 3.1 ficou da seguinte forma:

- valores positivos indicam que a câmera observa a frente da tela pela esquerda;
- valores negativos indicam que a câmera observa a frente tela pela direita;

Tabela 4.1: Relação de valores medidos e reais do eixo X da câmera em relação à tela

Gravação ( $i$ )	$\hat{X}_i$ (cm)	$X_i$ (cm)	$e_i^{(X)} =  X_i - \hat{X}_i $ (cm)
1	-0,14	0,00	0,14
2	-5,89	0,00	5,89
3	31,11	50,00	18,89
4	37,02	50,00	12,98
5	-27,22	-43,00	15,78
6	-6,26	0,00	6,26
7	2,03	0,00	2,03
8	49,01	50,00	0,99
9	37,51	40,00	2,49
10	-23,42	-40,00	16,58
11	-20,97	-40,00	19,03
12	27,85	35,00	7,15
13	0,07	0,00	0,07
14	-0,64	0,00	0,64
15	0,68	0,00	0,68
16	-19,85	-30,00	10,15
17	28,91	30,00	1,09
MAE <sup>(X)</sup>			7,78 cm

Para a Tabela 4.2, quanto maior a distância frontal entre a câmera e a tela, conforme os eixos de coordenadas da Figura 3.1, maior o valor positivo da medida.

Tabela 4.2: Relação de valores medidos e reais do eixo Y da câmera em relação à tela

Gravação ( $i$ )	$\hat{Y}_i$ (cm)	$Y_i$ (cm)	$e_i^{(Y)} =  Y_i - \hat{Y}_i $ (cm)
1	69,21	65,00	4,21
2	106,13	100,00	6,13
3	116,31	100,00	16,31
4	81,65	65,00	16,65
5	79,16	65,00	14,16
6	104,16	100,00	4,16
7	107,25	100,00	7,25
8	117,81	100,00	17,81
9	78,20	65,00	13,20
10	96,43	80,00	16,43
11	133,13	125,00	8,13
12	111,84	100,00	11,84
13	116,88	114,00	2,88
14	124,85	118,00	6,85
15	78,99	75,00	3,99
16	86,44	75,00	11,44
17	87,22	75,00	12,22
MAE <sup>(Y)</sup>			10,43 cm

Para a Tabela 4.3, a orientação utilizada no posicionamento vertical da câmera em relação à tela tomando os eixos de coordenadas da Figura 3.1 ficou da seguinte forma:

- valores positivos indicam que a câmera observa a tela de cima;
- valores negativos indicam que a câmera observa a tela de baixo;

Tabela 4.3: Relação de valores medidos e reais do eixo Z da câmera em relação à tela

Gravação ( $i$ )	$\hat{Z}_i$ (cm)	$Z_i$ (cm)	$e_i^{(Z)} =  Z_i - \hat{Z}_i $ (cm)
1	7,07	0,00	7,07
2	8,84	0,00	8,84
3	5,95	0,00	5,95
4	2,20	0,00	2,20
5	4,83	0,00	4,83
6	37,67	20,00	17,67
7	3,61	0,00	3,61
8	-0,20	0,00	0,20
9	-0,22	0,00	0,22
10	1,72	0,00	1,72
11	6,69	0,00	6,69
12	1,64	0,00	1,64
13	34,07	20,00	14,07
14	-26,52	-15,00	11,52
15	1,09	0,00	1,09
16	0,73	0,00	0,73
17	-0,20	0,00	0,20
MAE <sup>(Z)</sup>			6,16 cm

Como demonstração de análise, nota-se a partir das tabelas 4.1, 4.2 e 4.3 que  $(X_{11}, Y_{11}, Z_{11}) = (-40, 00; 125, 00; 0, 00)$ cm, enquanto  $(\hat{X}_{11}, \hat{Y}_{11}, \hat{Z}_{11}) = (-20, 97; 133, 13; 6, 69)$ cm. Comparando a Figura 4.3 com a Figura 4.4, é intuitivo perceber que os sistemas S2CP funcionaram para esta gravação, embora com erros de precisão.

O maior MAE entre todos os eixos é de 10,43 cm, para o eixo Y na Tabela 4.2. No entanto, a Figura 4.2 mostra, com a comparação entre  $A_i$  e  $\hat{A}_i$  para os três eixos, que os maiores erros neste eixo podem ocorrer devido aos seus maiores valores, pois está relacionado à distância frontal entre a câmera e a tela (ver a Figura 3.1), ou mesmo por desajustes na calibração, já que  $\hat{Y}_i$  é sempre maior que  $Y_i$ , o que pode indicar algum viés de estimativa.

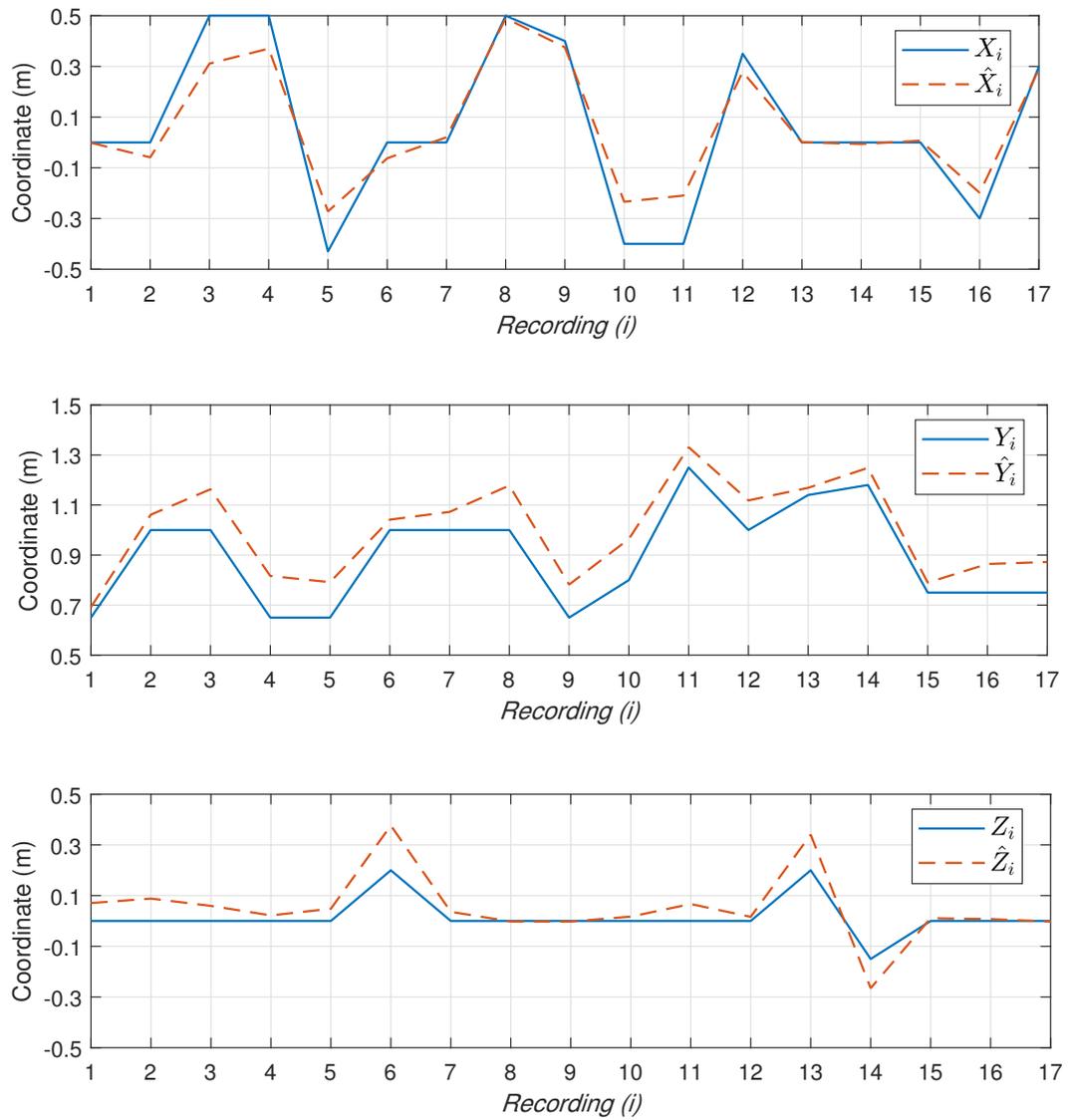


Figura 4.2: Comparação das coordenadas reais  $A_i = X_i, Y_i, Z_i$  (sólidas) e estimadas  $\hat{A}_i = X, Y, Z$  (tracejadas)

Vale destacar que esses erros de precisão não parecem ser outliers, mas sim seguir certos padrões. Por exemplo, observa-se melhor precisão ao longo do eixo  $Z$  ( $MAE^{(Z)} = 6,16$  cm), apesar de apresentar erros maiores quando  $Z_i$  aumenta (por exemplo,  $e_6^{(Z)} = 20$  cm). Por outro lado, o erro para essa gravação ( $i = 6$ ) levou a uma boa estimativa no eixo  $Y$  ( $e_6^{(Y)} = 4,16$  cm).

Na Tabela 4.4 encontram-se os valores das distâncias relativas entre a câmera e o centro do monitor utilizando os valores dos eixos  $X$ ,  $Y$  e  $Z$  apresentados nas Tabelas 4.1, 4.2 e 4.3 para calcular o vetor da distância euclidiana conforme a Equação 4.4 para a distância correta  $D$ , a Equação 4.5 para a distância estimada  $\hat{D}$  com os valores encontrados do posicionamento, e a diferença absoluta  $|D_i - \hat{D}_i|$ .

$$D = \sqrt{X^2 + Y^2 + Z^2} \quad (4.4)$$

$$\hat{D} = \sqrt{\hat{X}^2 + \hat{Y}^2 + \hat{Z}^2} \quad (4.5)$$

Tabela 4.4: Resultados da Distância Euclidiana Calculada

Gravação ( $i$ )	$\hat{D}_i$ (cm)	$D_i$ (cm)	$e_i^{(D)} =  D_i - \hat{D}_i $ (cm)
1	70,00	65,00	5,00
2	107,00	100,00	7,00
3	121,00	112,00	9,00
4	90,00	82,00	8,00
5	84,00	78,00	6,00
6	111,00	102,00	9,00
7	107,00	100,00	7,00
8	128,00	112,00	16,00
9	87,00	76,00	10,00
10	99,00	89,00	10,00
11	135,00	131,00	4,00
12	115,00	106,00	9,00
13	122,00	116,00	6,00
14	128,00	119,00	9,00
15	79,00	75,00	4,00
16	89,00	81,00	8,00
17	92,00	81,00	11,00

Tais dados de distância não são utilizados pelos algoritmos deste trabalho, porém servem como um indicador de acurácia dependente dos valores estimados dos três eixos, propagando quaisquer erros de acurácia. Nota-se que a distância euclidiana estimada foi maior que a correta para todas as gravações, mas apesar de apenas nas gravações 1, 11 e 15 a diferença ser menor que 5cm, quase todas as outras apresentam erro igual ou inferior a 10cm, exceto pelas gravações 8 e 17. De um modo geral, observa-se que o eixo Y possui maior impacto na acurácia da distância euclidiana calculada, visto que apresenta os maiores valores dentre os três eixos. Para ambas as gravações 8 e 17, ocorreram os maiores erros no eixo Y.

## 4.2 Cálculo de distâncias entre câmera e tela

Nesta Seção são apresentadas as distâncias calculadas através do método apresentado na Seção 3.7, onde a Equação 3.3 depende da calibração de uma constante  $C$  em função do ângulo de visão da lente da câmera.

Na Tabela 4.5 estão os valores das distâncias relativas entre a câmera e o centro do monitor detectada com um coeficiente  $C = 1,60213$ , encontrado para uma distância conhecida  $d=100\text{cm}$  entre a câmera e a tela. A distância correta é  $D$ , a distância calculada é  $\hat{D}$  e a diferença absoluta é  $|D_i - \hat{D}_i|$ .

Tabela 4.5: Relação de valores de distância detectada com a mensagem codificada e processamento dos quadros, e a distância correta, para  $C = 1,60213$

Gravação ( $i$ )	$\hat{D}_i$ (cm)	$D_i$ (cm)	$e_i^{(D)} =  D_i - \hat{D}_i $ (cm)
1	69,00	65,00	4,00
2	104,00	100,00	4,00
3	103,00	112,00	8,00
4	68,00	82,00	14,00
5	71,00	78,00	7,00
6	109,00	102,00	7,00
7	104,00	100,00	4,00
8	108,00	112,00	4,00
9	68,00	76,00	8,00
10	87,00	89,00	2,00
11	128,00	131,00	3,00
12	107,00	106,00	1,00
13	119,00	116,00	4,00
14	122,00	119,00	3,00
15	79,00	75,00	4,00
16	79,00	81,00	2,00
17	79,00	81,00	2,00

Para analisar o impacto de alterar/recalibrar o valor da constante  $C$  no cálculo da distância, na Tabela 4.6 encontram-se as distâncias detectadas para  $C = 1,51936$ , calibrada com uma distância conhecida  $d=75\text{cm}$  entre a câmera e a tela.

Tabela 4.6: Relação de valores de distância detectada com a mensagem codificada e processamento dos quadros, e a distância correta, para  $C = 1,51936$

Gravação ( $i$ )	$\hat{D}_i$ (cm)	$D_i$ (cm)	$e_i^{(D)} =  D_i - \hat{D}_i $ (cm)
1	65,00	65,00	0,00
2	99,00	100,00	1,00
3	98,00	112,00	14,00
4	64,00	82,00	18,00
5	68,00	78,00	10,00
6	103,00	102,00	1,00
7	99,00	100,00	1,00
8	102,00	112,00	9,00
9	64,00	76,00	12,00
10	83,00	89,00	6,00
11	121,00	131,00	10,00
12	102,00	106,00	4,00
13	113,00	116,00	2,00
14	116,00	119,00	3,00
15	75,00	75,00	0,00
16	75,00	81,00	5,00
17	75,00	81,00	5,00

Apesar de ambas as calibrações da constante  $C$  apresentarem resultados satisfatórios para a distância calculada, observa-se que  $C = 1,60213$  apresentou melhor exatidão para a câmera e configurações utilizadas com o POCO F5. Ou seja, para este experimento, calibrar  $C$  com uma distância conhecida de 100cm foi mais eficaz ao invés de uma distância de 75cm. Para maiores conclusões sobre qual a distância mais recomendada para a calibração de  $C$ , poderia ser conduzido um experimento maior no futuro. Novamente, vale ressaltar que tal constante precisaria ser recalibrada manualmente para diferentes câmeras, e seria necessário conduzir um novo processo de comparação entre calibrações para determinar a melhor.

Assim como as distâncias Euclidianas apresentadas na Tab. 4.4, esta informação da distância calculada através da Eq. 3.3 não foi utilizada pelos demais processamentos deste trabalho, e servem apenas como um indicador de análise de acurácia da Equação 3.3 aplicada em condições reais. Ao comparar os resultados das Tabelas 4.5 e 4.6 com os da Tabela 4.4, nota-se que utilizar o método com a Equação 3.3 apresentou mais resultados com erros abaixo de 5cm do que o método de distância euclidiana, o que pode sugerir a preferência de escolher o método da Seção 3.7 caso em futuras aplicações este seja um dado necessário.

### 4.3 Identificação e validação visual

Na Figura 4.3 é exibido um quadro da gravação 11 com as quatro quinas da tela e os pontos intermediários identificados por pontos verdes. Nos gráficos 3D exibidos na Figura 4.4, o ponto representa o centro da câmera, e o paralelepípedo vermelho representa a tela em escala 1:1 com o restante do gráfico. A posição do ponto da câmera condiz com os valores X, Y e Z encontrados na gravação 11 para cada eixo na coluna 'Distância detectada' nas Tabelas 4.1, 4.2 e 4.3.

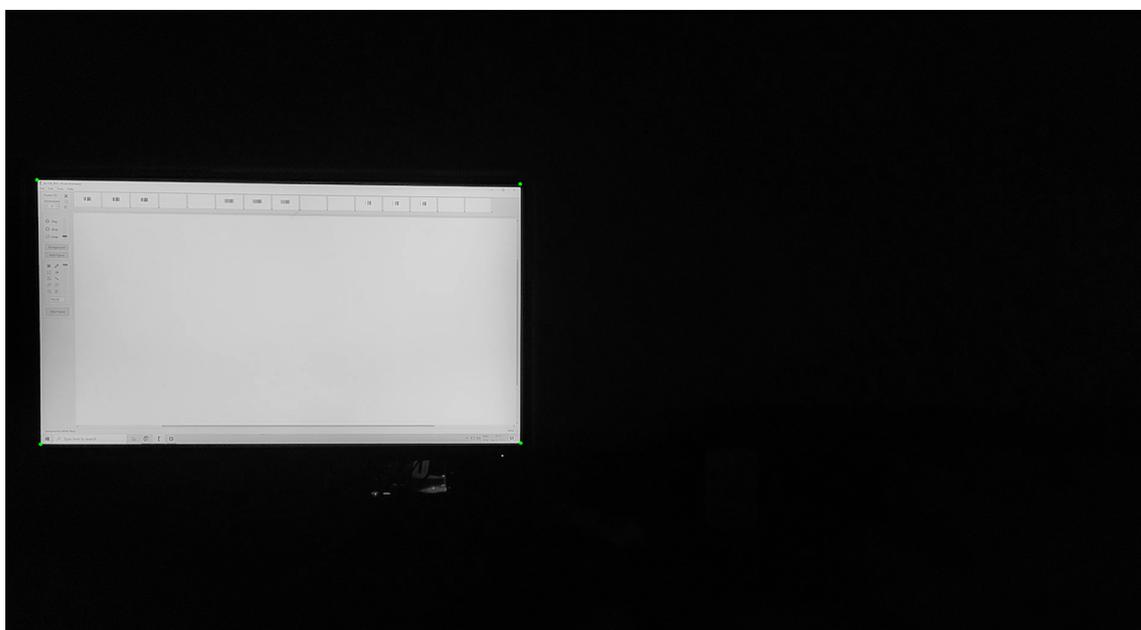


Figura 4.3: Quadro da gravação 11. Os pontos verdes indicam a identificação das quatro quinas do monitor

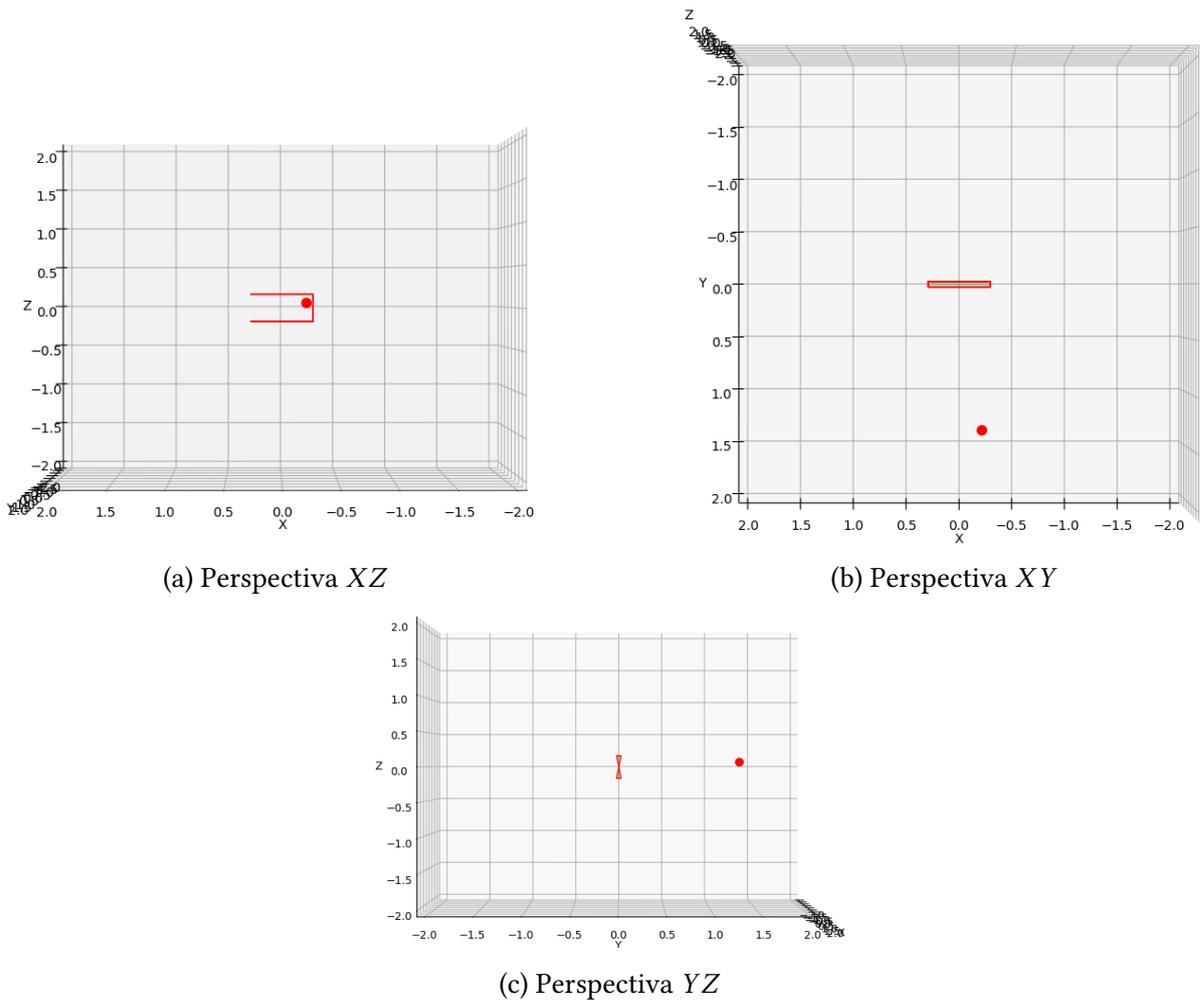


Figura 4.4: Posição plotada da câmera para a gravação 11 com diferentes perspectivas do gráfico 3D em (a), (b) e (c). O ponto vermelho representa a câmera e o paralelepípedo representa a tela.

## 4.4 Precisão e medidas de erros

Como este artigo apresenta o primeiro esquema S2C para posicionamento, é possível comparar os resultados de MAE da Seção 4.1 com o trabalho mais próximo de (FENG et al., 2023), que relatou um MAE de 6,5 cm para distâncias euclidianas semelhantes. No entanto, o desempenho aparentemente superior de (FENG et al., 2023) pode ser atribuído ao uso de uma fonte de luz estacionária montada no teto com gravações de código de barras foscas, juntamente com o auxílio de um sensor de ângulo embutido no smartphone para seu "posicionamento pseudo de lâmpada dupla". Em outras palavras, (FENG et al., 2023) variou apenas dois eixos, usou uma fonte de luz forte e contou com hardware adicional do receptor (por exemplo, sensor de ângulo). Em contraste, nosso sistema operou em condições de baixa luminosidade, onde a fonte de luz primária (a tela) pode introduzir variações sutis de contraste e brilho inerentes à exibição, potencialmente degradando a precisão da detecção de cantos e da estimativa de pose 3D. No entanto, de modo geral, a precisão de posicionamento do SC2P é ligeiramente inferior à de (FENG et al., 2023), mas ainda superior aos valores relatados por (HAO; CHEN, J.; WANG, 2019), onde os valores de MAE variam de 13,1 cm ou mais.

Finalmente, outras fontes de erros podem incluir pequenos desalinhamentos de medições, mesmo após o ajuste manual com o editor de perspectivas (Apêndice A), imprecisões intrínsecas na solução do *P4P* ou até mesmo erros de arredondamento numérico dos algoritmos utilizados, incluindo aqueles das rotinas do OpenCV, que podem ter contribuído para os erros de estimativa.

Outro fator que possivelmente impacta a precisão do nosso sistema é a matriz de pré-calibração, que é derivada de gravações reais para estimar uma correção eficaz de distorção. Aperfeiçoar a etapa de calibração, que pode melhorar a correção de distorção, provavelmente aumentará ainda mais a precisão.

## Capítulo 5

# Adaptação do S2CP para Protocolo Tensorial

A principal modificação implementada após o desenvolvimento do esquema do Capítulo 3 consistiu na substituição do uso de códigos de barras BC-128 por um protocolo de comunicação S2C próprio, desenvolvido pelo Prof. Leandro R. Ximenes et. al., baseado na modelagem tensorial dos pixels codificados por símbolos de informação.

O esquema, primeiramente apresentado em (XIMENES; LAREDO; ARTHUR, 2023), e depois aprimorado em outros trabalhos (XIMENES; ALVES, 2023; XIMENES; GONÇALVES et al., 2024; XIMENES; CARDOSO, 2024), é baseado na codificação espaço-temporal de um bloco de quadros de vídeo de dimensões  $M \times N$  do lado do transmissor em um tensor de dimensões  $MN \times F \times K$  no receptor, em que  $F$  é o número de blocos utilizados e  $K$  é o número de símbolos transmitidos por bloco.

Também no receptor, um dos vários algoritmos propostos nos trabalhos deste parágrafo podem decompor esse tensor em componentes de interesse, como a matriz de símbolos e a sequência de vídeo original sem codificação. A Figura 5.1 ilustra a geração do tensor a partir do bloco de quadros capturados pela câmera no lado do receptor.

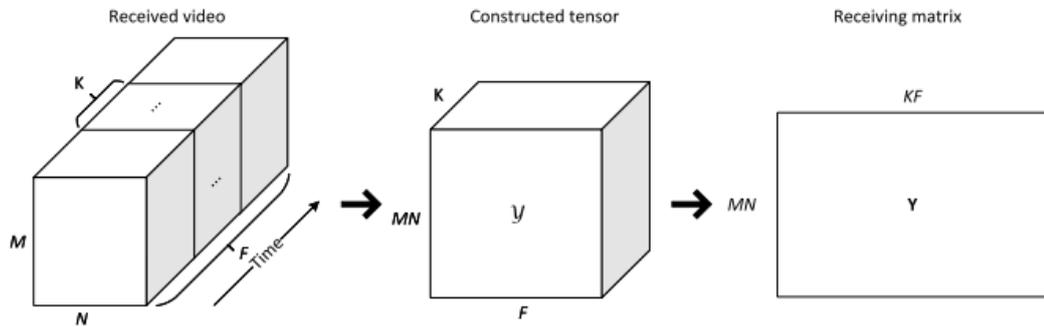


Figura 5.1: Organização dos quadros de vídeo capturados com o protocolo tensorial. O vídeo recebido é composto por um bloco de  $K \times F$  quadros. (XIMENES; LAREDO; ARTHUR, 2023)

Em comparação ao código de barras, o esquema tensorial apresenta as seguintes vantagens:

1. a codificação de vídeos arbitrários junto à mensagem binária transmitida, de forma a tornar a transmissão da informação de posicionamento imperceptível ao olho humano e cumprir com objetivo primário de qualquer tela, a de transmitir imagens antes de quaisquer informações posicionais;
2. a possibilidade de efetuar a correção da degradação (e.g., borramento) de forma intrínseca, sem a necessidade de uma etapa de calibração precedente;
3. compromissos (*trade-offs*) entre ganhos de taxas de dados e ganhos de diversidade (e.g. robustez à ruídos), não presentes em códigos de barras.

Embora relevantes, estas três vantagens não são exploradas neste documento. O intuito aqui é de demonstrar que este esquema, novo na literatura, é capaz de ser utilizado também para o esquema de posicionamento S2CP proposto.

Tomando como referência o fluxograma da Figura 3.2 de funcionamento do S2CP apresentado no início do Capítulo 3, esta adaptação consistiu na alteração dos algoritmos empregados nos blocos 1, 5 e 6 somente, conforme ilustrado na Figura 5.2. Estes blocos serão apresentados nas Seções 5.1, 5.2 e 5.3, respectivamente.

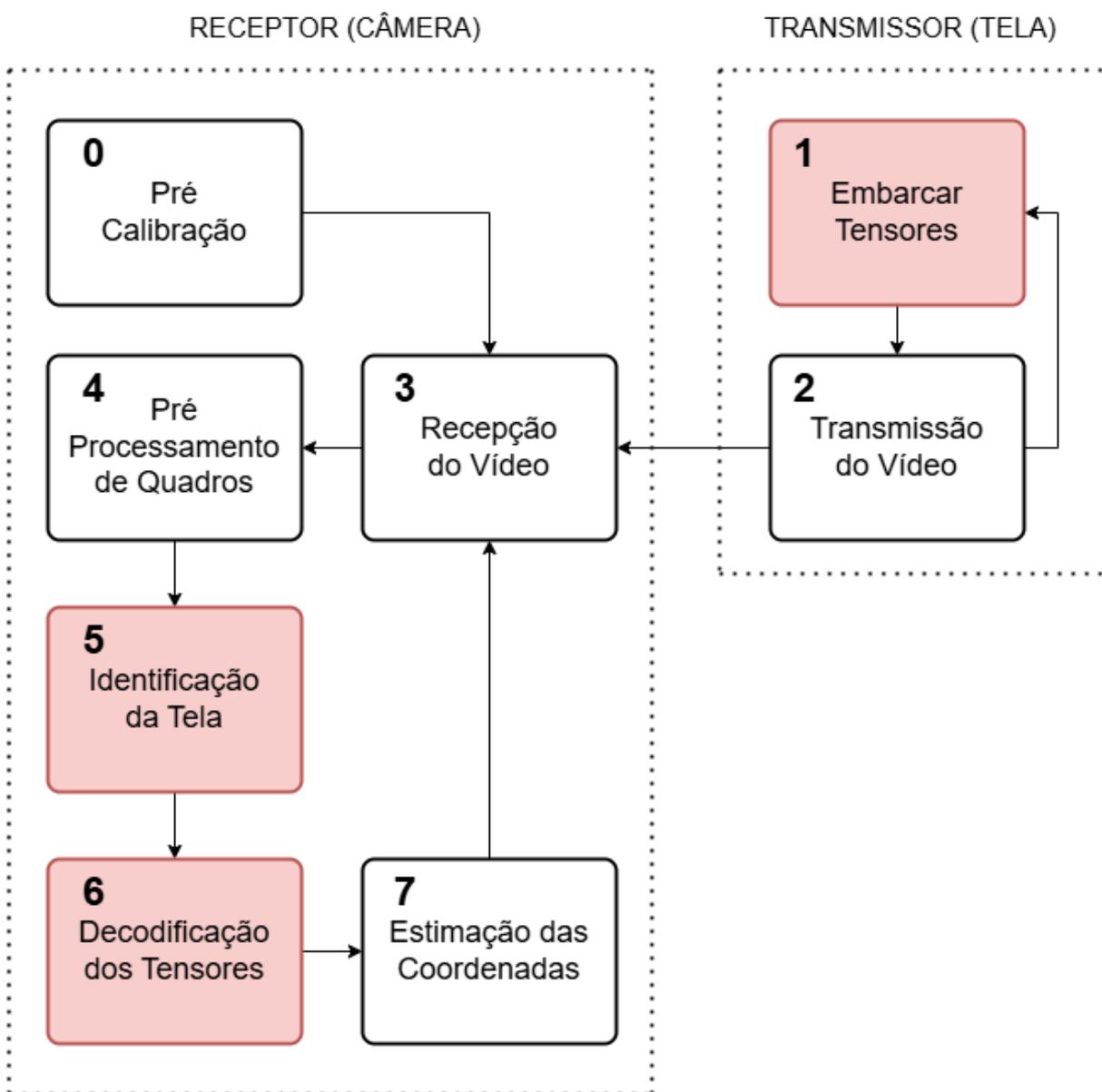


Figura 5.2: Fluxograma adaptado para o protocolo tensorial, com ênfase nos blocos onde esta adaptação modificou o sistema original.

## 5.1 Codificação da mensagem segundo esquema tensorial

Esta Seção aborda a alteração do bloco 1 da Figura 5.2 para inserir a codificação tensorial no lugar do código de barras. Por uma questão de brevidade, a codificação usou a mesma estrutura do trabalho em (XIMENES; LAREDO; ARTHUR, 2023), além do mesmo decodificador. Além disso, ao substituir o código de barras, foi proposto uma nova estrutura de organização de dados, mais completa do que aquela utilizada nos capítulos anteriores, indicada na Figura 5.3.

Conforme o esboço da estrutura de dados serializados do protocolo na Figura 5.3, ao montar a sequência de mensagens no novo protocolo, buscou-se ter a menor quantidade possível de

quadros de cabeçalho e sincronia para garantir maior flexibilidade na quantidade de dados transmitidos pertinentes ao posicionamento e outras informações para explorar o potencial dessa nova tecnologia proposta de S2CP.

De forma geral, o quadro serial de bits apresentado na Figura 5.3 é mapeado/codificado horizontalmente e depois verticalmente nos pixels do quadro original. A modulação usada é binária do tipo OOK, em que o *bit* 1 corresponde a um pixel aceso (brilho máximo), e o *bit* 0 a um pixel apagado (brilho mínimo).

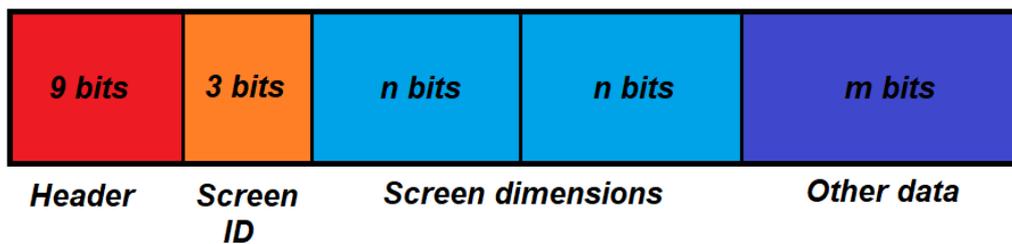


Figura 5.3: Esboço do protocolo tensorial em desenvolvimento pelo LPS.

Por tratar-se de um protocolo atualmente em desenvolvimento, a quantidade de *bits* utilizados para as dimensões e para dados adicionais não foi definida com valores absolutos. A princípio foi adotado  $n = 12$ , permitindo até  $2^{12} = 4096$  mm, ou seja, representar telas de até  $4 \times 4$  metros, com resolução de 1 mm. O *header* deve ser uma mensagem reservada. Para estes testes, utilizou-se (111111110). O *Screen ID* permite a identificação de até  $2^3 = 8$  telas, porém foi utilizada apenas uma. Por fim, *Other data* foram transmitidos como *bits* aleatórios com distribuição uniforme entre 0 e 1.

Após a conversão dos dados serializados à tela com o uso da codificação tensorial, conforme a Figura 5.4, cada quadro de dimensão  $J \times L$  é dividido em  $V \times H$  blocos. Cada bloco tem dimensões as  $M \times N$  apresentadas na Figura 5.1. Neste caso, há um total de  $MV = J$  pixels verticais e  $HN = L$  pixels horizontais. Observa-se que cada bloco pode transmitir informações diferentes para ganho de taxa de transmissão, ou informações redundantes para maior robustez às degradações do canal de transmissão. Neste protótipo, foi optado por transmitir dados redundantes para prover ganhos de diversidade. (XIMENES; GONÇALVES et al., 2024)

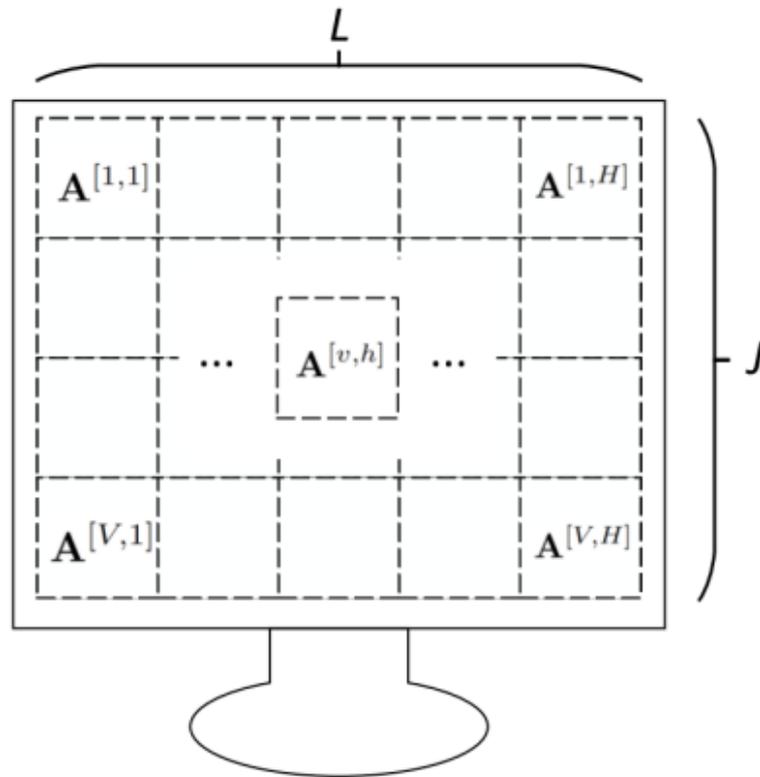


Figura 5.4: Exemplo de um quadro codificado após conversão da mensagem serializada para o protocolo tensorial. (XIMENES; CARDOSO, 2024)

## 5.2 Identificação da Tela por Variação Temporal

Esta Seção aborda a alteração do bloco 5 da Figura 5.2, relacionado à identificação da ROI referente à tela pelo receptor. O funcionamento do novo algoritmo que será apresentado está ilustrado no fluxograma da Figura 5.5.

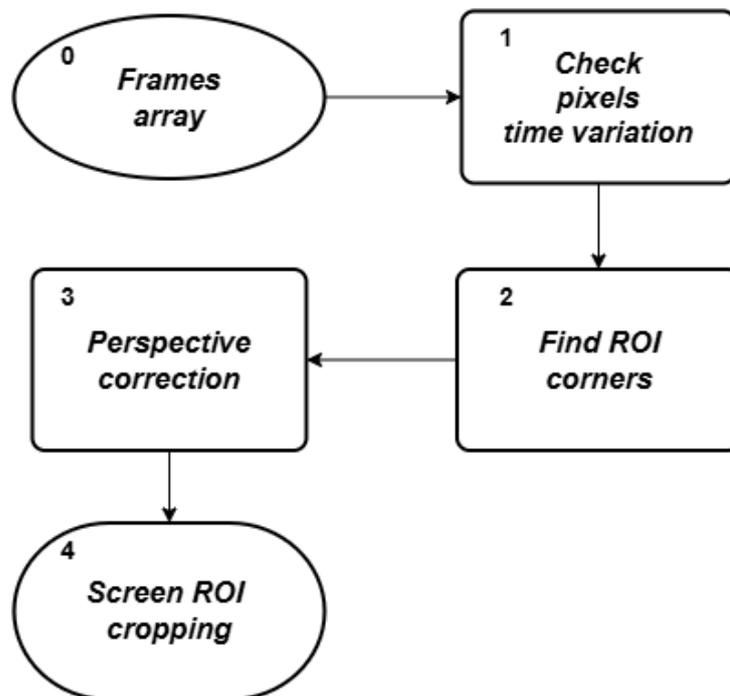


Figura 5.5: Fluxograma da análise de variação de pixels.

Anteriormente, na Seção 3.5 do Capítulo 3, a identificação da tela era realizada por um sistema de pontuação baseado nos elementos internos das ROIs candidatas a ser uma tela. Desta vez, a maneira adotada para identificar a região onde o código está sendo transmitido foi a análise de variação temporal de pixels dentro das regiões de interesse candidatas. A ideia é que ao contrário de padrões estatísticos de códigos de barras ou *QR Codes*, a comunicação S2C transmite informações diferentes quadro a quadro. Portanto, a própria variação temporal é um bom indicador – e de baixa complexidade computacional – da ROI.

Toma-se uma quantidade suficiente de quadros (bloco 0 Figura 5.5) para identificação da ROI que contém uma variação temporal de pixels dentro de limites de variação (bloco 1 Figura 5.5). Tais limites servem para desconsiderar regiões com pouca variação, ou regiões de altíssima variação, ao ponto de caracterizarem-se como ruídos aleatórios.

Uma vez localizada a região dos quadros com variação adequada, são identificados os quatro vértices da ROI da tela (bloco 2 Figura 5.5), que futuramente também serão usados como os *Image Points* para o posicionamento. Em seguida, aplica-se uma homografia tomando essas quinas como pontos a serem corrigidos para centralizar e alinhar a tela nos quadros (bloco 3 Figura 5.5). Após essa transformação de alinhamento, os *frames* são recortados e redimensionados para apenas a nova ROI corrigida da tela (bloco 4 Figura 5.5).

Para os testes iniciais, as simulações foram modeladas com a câmera e a tela fixas, onde foi desenhada uma paisagem simples composta apenas por uma região retangular branca com uma moldura preta fina, e um plano de fundo cinza, conforme a Figura 5.6. Com essas tonalidades escolhidas, ao aplicar *thresholding* e identificação de contornos, apenas a tela fica destacada. Dessa forma, ao passar pelo pré processamento de quadros, a tela sempre é identificada corretamente.

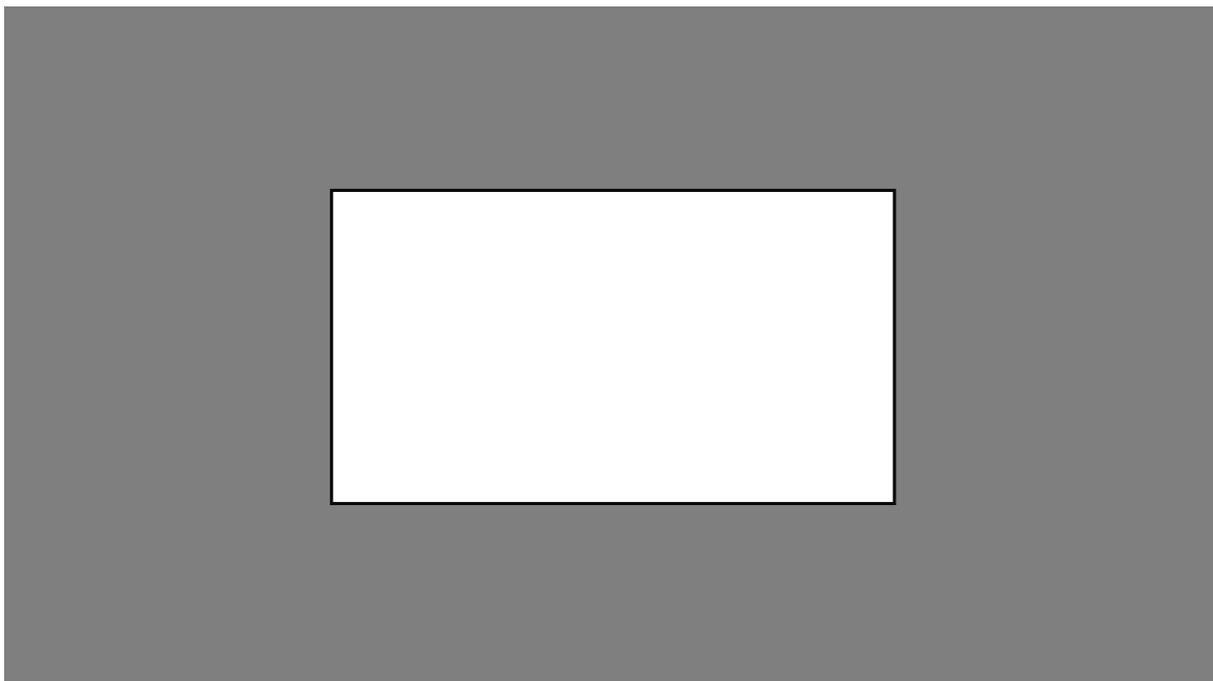


Figura 5.6: Cena desenhada para simulações. A região que representa a tela está posicionada a 1 metro de distância centralizada na câmera, com as mesmas dimensões do monitor utilizado no Cap. 3.

Com esse quadro artificial, é possível gerar vídeos com editores de vídeo como o *Camtasia Studio*, inserindo qualquer protocolo de comunicação por cima da tela. Garantiu-se que os vídeos foram gerados no formato AVI sem perdas de compressão, com dimensões 1920x1080p e 60FPS. Para gerar diferentes perspectivas, basta aplicar homografia com o editor de perspectiva apresentado no Apêndice A.

Para simular o cenário de encontrar corretamente a tela, foram inseridas outras regiões retangulares que podem ser sinalizadas como possíveis *displays*, conforme a Figura 5.7. Das três regiões candidatas, apenas a da tela contém variações significativas nos pixels ao longo do tempo, pois as outras (que podem ser janelas, quadros, entre outras coisas) possuem conteúdo interno estático.

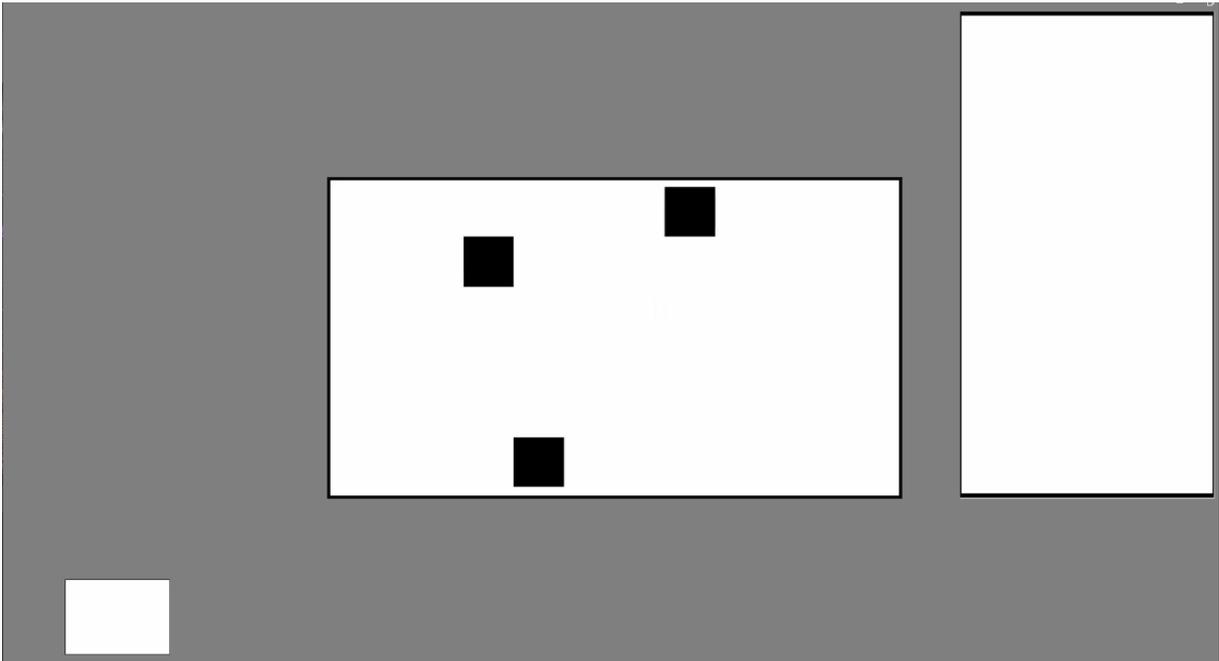


Figura 5.7: Cena simulada com múltiplas telas. Somente a tela central contém os símbolos matriciais sendo transmitidos ao longo do tempo, enquanto que as outras ficam em branco durante a simulação.

O algoritmo de identificação do protocolo tensorial através da análise de variação temporal dos pixels foi capaz de identificar corretamente a ROI da tela após a coleta e cálculo da variância dos pixels ao longo de  $F$  quadros. Para um bloco de vídeo capturado  $X$  com  $F$  quadros, a variância temporal do pixel  $(i, j)$  é dada por:

$$\sigma^2(i, j) = \sum_{k=1}^F (X(i, j, k) - \mu(i, j))^2 \quad (5.1)$$

$$\mu(i, j) = \sum_{k=1}^F X(i, j, k) \quad (5.2)$$

Logo, é considerada a ROI com variação temporal adequada onde há a maior quantidade de pixels  $(i, j)$  nos quais  $\sigma^2(i, j) \geq \gamma$ , em que  $\gamma > 0$  define o limite se o pixel variou ou não ao longo do tempo. Nota-se que este limiar de variação pode ser escolhido de forma empírica.

Dessa forma já foi possível identificar se houve um padrão temporal em partes dos dados transmitidos, visto que os quadros de sincronia e dimensões da tela possuem símbolos matriciais com informações cíclicas.

Conforme outro exemplo da Figura 5.8, onde foi utilizado o editor de perspectivas para simular outro posicionamento da câmera, a tela com a mensagem sendo transmitida foi detectada corretamente ao invés da tela em branco após a análise de variação temporal.

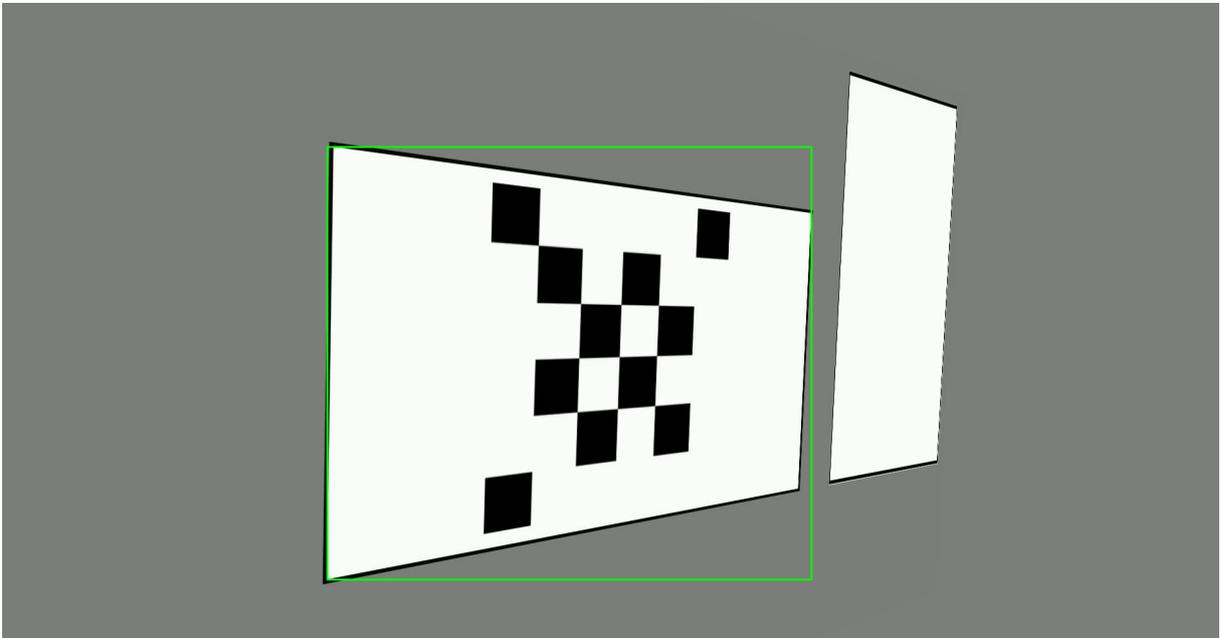


Figura 5.8: Tela detectada por variação temporal de pixels.

Para o restante dos processamentos da mensagem, foi aplicada uma homografia para deixar a ROI da tela alinhada de maneira frontal, e recortados os quadros do vídeo somente nessa nova ROI alinhada, conforme o exemplo da Figura 5.9.

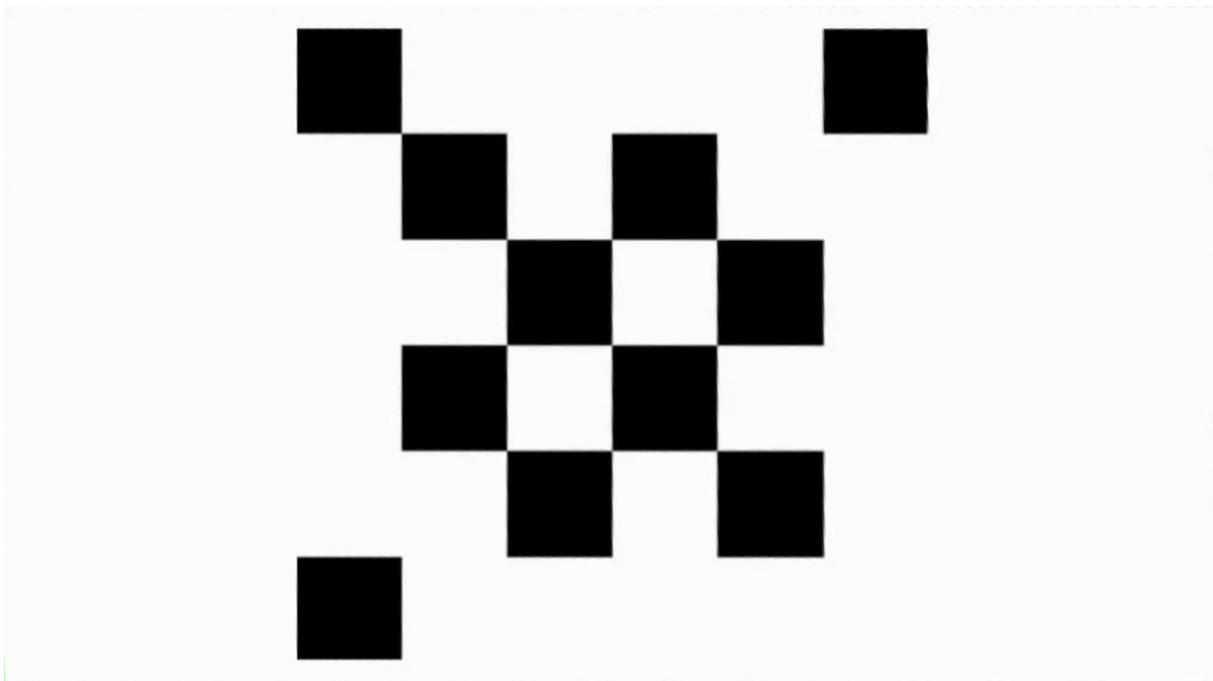


Figura 5.9: Homografia e recorte da tela detectada, permitindo que os demais processamentos recebam o conteúdo da tela sem distorções.

Conforme observado na Figura 5.10, onde contém um *log* do *console* de execução do Pycharm durante a execução do sistema, esse processo de identificação isola a ROI da tela que contém os dados transmitidos. Para as simulações e gravações realizadas neste teste, foram utilizados com sucesso limiares de variância mínimo e máximo de 700000 e 1000000, ou seja, a variação de pixels entre quadros só era considerada válida nesse intervalo. Valores de variação abaixo do limiar foram considerados dados estáticos, e valores acima como ruídos aleatórios.

```
Relevant variance for screen at (x=495, y=269, w=932, h=526).
```

Figura 5.10: *Console* com informação da região de interesse detectada. As coordenadas  $x$  e  $y$  indicam a posição da quina superior esquerda da tela, enquanto que as dimensões  $w$  e  $h$  indicam a largura e altura dessa ROI.

Foram realizados mais 4 testes, apenas alterando a quantidade de telas em branco entre zero e duas, para validar o algoritmo de identificação da tela correta através da análise de variação de pixels. Em todas o sistema identificou a ROI da tela corretamente.

### 5.3 Identificação e decodificação do Tensor

Esta Seção aborda a alteração do bloco 6 da Figura 5.2. Com os novos quadros contendo apenas a ROI da tela identificada e perspectiva corrigida, resultantes da homografia e recorte demonstrados na Seção 5.2, foi desenvolvido um código que realiza a identificação de formas quadradas ou retangulares nos quadros a partir da detecção de contornos, conforme o fluxograma na Figura 5.11. O intuito deste processo agora é recortar uma nova ROI correspondente somente à mensagem transmitida.

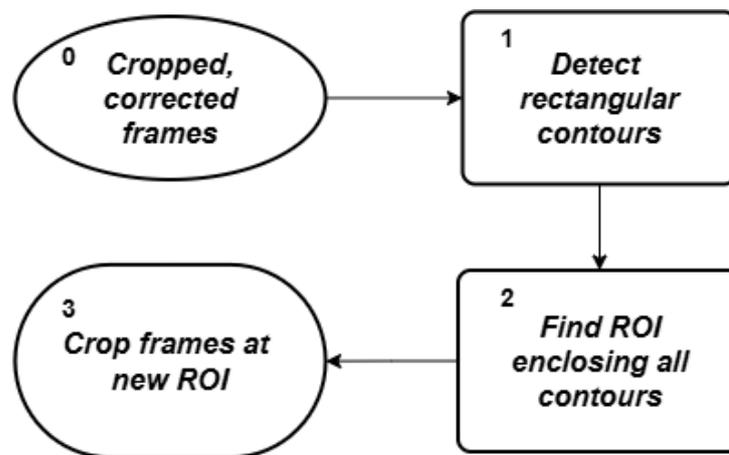


Figura 5.11: Fluxograma da detecção das matrizes codificadas.

São analisadas as formas geométricas desses contornos para verificar se possuem características compatíveis com quadrados ou retângulos (bloco 1). Durante o processo, é calculado o centroide de cada contorno identificado como quadrangular e uma lista de coordenadas centrais é gerada. Simultaneamente, a função determina a menor ROI que engloba todos os contornos detectados, calculando suas dimensões mínimas e máximas (bloco 2). Ao final, o código retorna os detalhes das formas identificadas e a delimitação espacial geral. Com esses dados, é realizado um novo recorte apenas da região que contém os dados matriciais (bloco 3).

Foi realizada uma gravação com uma perspectiva aplicada artificialmente através do editor de perspectivas. Tal exemplo pode ser observado na Figura 5.12.

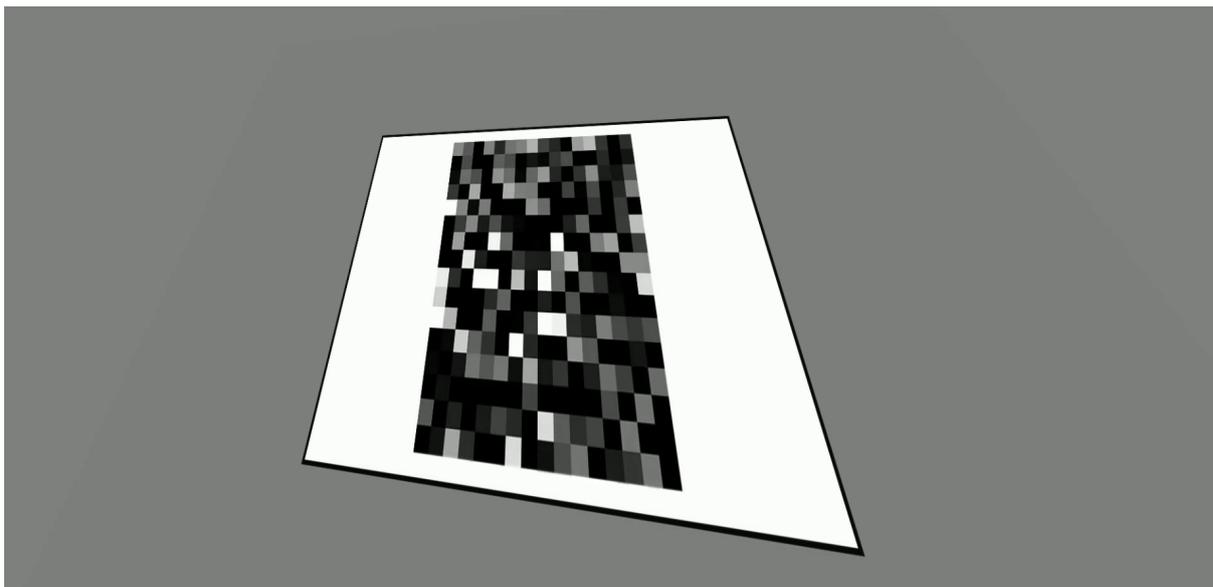


Figura 5.12: Simulação do protocolo tensorial com perspectiva aplicada.

Mesmo sob essa forte aplicação de perspectiva, observa-se na Figura 5.13 a correta identificação e ajuste de perspectiva dos tensores para o recorte a ser utilizado na decodificação dos dados transmitidos. É possível notar também pequenos artefatos e perda de resolução nas extremidades da matriz que estavam menores devido a perspectiva aplicada (quadrante superior esquerdo), porém ainda é possível distinguir todos os seus pontos.

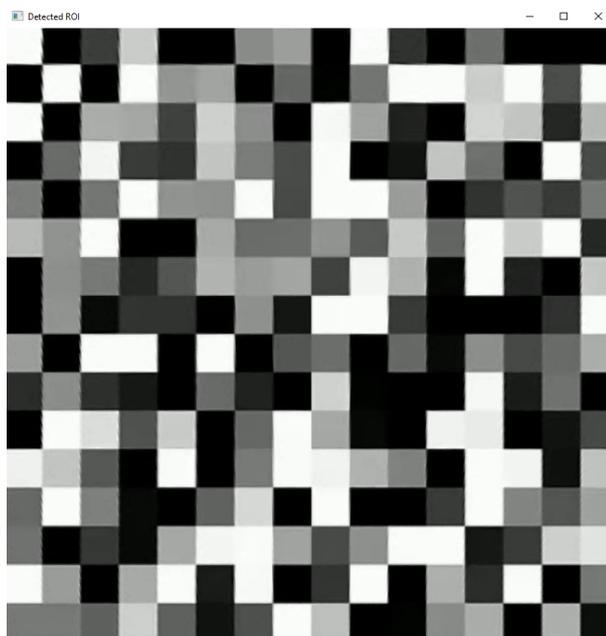


Figura 5.13: Recorte de um quadro do tensor após ajuste de perspectiva. Após o recorte dessa região de interesse, os demais quadros do vídeo também foram recortados e salvos até completar um ciclo de transmissão.

Com o vídeo recortado dessa forma, o decodificador também em desenvolvimento em (XIMENES; LAREDO; ARTHUR, 2023) é capaz de processar os quadros e decodificar o tensor a fim de extrair as informações das dimensões da tela, conforme o exemplo ilustrado na Figura 5.14, referente à gravação 7 da Seção 5.4.

```
--- Decodificação bloco a bloco ---  
  
SER Médio: 0.001736  
BER: 0.006944  
Header decodificado: 1 1 1 1 1 1 1 1 0  
ID decodificado: 0 0 1  
Dimensão Vertical: 330  
Dimensão Horizontal: 565
```

Figura 5.14: Exemplo de dados retornados pelo decodificador do protocolo tensorial. As dimensões encontradas condizem corretamente com as dimensões físicas do monitor Ozone®OZDSP25PRO utilizado.

## 5.4 Testes reais com o protocolo tensorial

Assim como os experimentos gravados demonstrados na Figura 4.1, para estes testes com o protocolo tensorial, é possível visualizar a estrutura e disposição dos materiais utilizados na Figura 5.15.



Figura 5.15: Demonstração do ambiente de gravação. O *smartphone* está fixado a um tripé ajustável para poder ser posicionado adequadamente em relação a tela transmissora dos códigos tensoriais.

As Tabelas 5.1, 5.2 e 5.3 apresentam a comparação entre os valores estimados pelo sistema de posicionamento e os valores corretos das distâncias utilizadas, de maneira similar às Tabelas 4.1, 4.2 e 4.3 da Seção 4.1. Sendo assim, novamente são mapeadas as coordenadas reais  $A = \{X, Y, Z\}$ , as estimativas calculadas  $\hat{A}_i$ , e o erro absoluto  $e_i^{(A)}$  entre os valores corretos e calculados.

A coluna **Modo** refere-se às dimensões utilizadas na codificação da mensagem com o protocolo tensorial. Conforme as definições apresentadas com as Figuras 5.1 e 5.4, foram testados os quatro **Modos** seguintes:

- JL4\_MN2\_F5: quadro de dimensões totais  $4 \times 4$ , com subblocos de dimensões  $2 \times 2$ , e  $F = 5$  quadros;

- JL8\_MN2\_F5: quadro de dimensões totais  $8 \times 8$ , com subblocos de dimensões  $2 \times 2$ , e  $F = 5$  quadros;
- JL8\_MN4\_F5: quadro de dimensões totais  $8 \times 8$ , com subblocos de dimensões  $4 \times 4$ , e  $F = 5$  quadros;
- JL8\_MN4\_F10: quadro de dimensões totais  $8 \times 8$ , com subblocos de dimensões  $4 \times 4$ , e  $F = 10$  quadros.

Para o cálculo do **MAE**, foi utilizada novamente a Equação 4.3, porém agora para  $N = 11$  gravações.

Foram utilizadas somente três perspectivas ao longo destes testes, repetindo-se cada perspectiva para cada **MAE**, da seguinte forma:

- A: gravação centralizada e alinhada verticalmente com a tela, a 75cm de distância ( $X=0$ ,  $Y=75$ ,  $Z=0$ );
- B: gravação centralizada e alinhada verticalmente com a tela, a 100cm de distância ( $X=0$ ,  $Y=100$ ,  $Z=0$ );
- C: gravação deslocada à esquerda em 70 cm, a 80cm de distância ( $X=70$ ,  $Y=80$ ,  $Z=0$ );

Tabela 5.1: Relação de valores medidos e reais do eixo X da câmera em relação à tela com o protocolo tensorial

Gravação ( $i$ )	Modo	Perspectiva	$\hat{X}_i$ (cm)	$X_i$ (cm)	$e_i^{(X)} =  X_i - \hat{X}_i $ (cm)
1	JL4_MN2_F5	A	0,75	0,00	0,75
2	JL4_MN2_F5	B	-0,05	0,00	0,05
3	JL4_MN2_F5	C	72,99	70,00	2,99
4	JL8_MN2_F5	A	0,85	0,00	0,85
5	JL8_MN2_F5	C	72,96	70,00	2,96
6	JL8_MN4_F5	A	0,89	0,00	0,89
7	JL8_MN4_F5	B	0,00	0,00	0,00
8	JL8_MN4_F5	C	72,32	70,00	2,32
9	JL8_MN4_F10	A	0,83	0,00	0,83
10	JL8_MN4_F10	B	-0,16	0,00	0,16
11	JL8_MN4_F10	C	72,18	70,00	2,18
MAE <sup>(X)</sup>			1,27 cm		

Para a Tabela 5.2, quanto maior a distância frontal entre a câmera e a tela, conforme os eixos de coordenadas da Figura 3.1, maior o valor positivo da medida.

Tabela 5.2: Relação de valores medidos e reais do eixo Y da câmera em relação à tela com o protocolo tensorial

Gravação ( $i$ )	Modo	Perspectiva	$\hat{Y}_i$ (cm)	$Y_i$ (cm)	$e_i^{(Y)} =  Y_i - \hat{Y}_i $ (cm)
1	JL4_MN2_F5	A	80,06	75,00	5,06
2	JL4_MN2_F5	B	106,60	100,00	6,60
3	JL4_MN2_F5	C	91,98	80,00	11,98
4	JL8_MN2_F5	A	79,99	75,00	4,99
5	JL8_MN2_F5	C	91,60	80,00	11,60
6	JL8_MN4_F5	A	79,99	75,00	4,99
7	JL8_MN4_F5	B	106,40	100,00	6,40
8	JL8_MN4_F5	C	91,91	80,00	11,91
9	JL8_MN4_F10	A	79,96	75,00	4,96
10	JL8_MN4_F10	B	106,27	100,00	6,27
11	JL8_MN4_F10	C	92,08	80,00	12,08
MAE <sup>(Y)</sup>			7,89 cm		

Para a Tabela 5.3, a orientação utilizada no posicionamento vertical segue a do eixo Z da Figura 3.1.

Tabela 5.3: Relação de valores medidos e reais do eixo Z da câmera em relação à tela com o protocolo tensorial

Gravação ( $i$ )	Modo	Perspectiva	$\hat{Z}_i$ (cm)	$Z_i$ (cm)	$e_i^{(Z)} =  Z_i - \hat{Z}_i $ (cm)
1	JL4_MN2_F5	A	1,55	0,00	1,55
2	JL4_MN2_F5	B	5,08	0,00	5,08
3	JL4_MN2_F5	C	1,63	0,00	1,63
4	JL8_MN2_F5	A	2,02	0,00	2,02
5	JL8_MN2_F5	C	1,72	0,00	1,72
6	JL8_MN4_F5	A	2,17	0,00	2,17
7	JL8_MN4_F5	B	5,52	0,00	5,52
8	JL8_MN4_F5	C	1,75	0,00	1,75
9	JL8_MN4_F10	A	1,84	0,00	1,84
10	JL8_MN4_F10	B	3,27	0,00	3,27
11	JL8_MN4_F10	C	1,64	0,00	1,64
MAE <sup>(Z)</sup>			2,56 cm		

A gravação com perspectiva B para o **Modo** JL8\_MN2\_F5 não resultou em uma identificação correta da ROI da mensagem transmitida, por isso não há tal perspectiva listada nas tabelas. Tal erro indica que ainda há a possibilidade (e necessidade) de refinamento do método de identificação dessa ROI.

Ao comparar os erros de posicionamento e **MAEs** obtidos com os da Seção 4.1, observa-se uma melhora na acurácia do posicionamento. Tal noção, entretanto, deve-se apenas ao fato que as gravações realizadas para validar o sistema S2CP com o protocolo tensorial foram feitas em maior parte de forma centralizada com a tela, sem variação vertical do eixo Z e com a distância máxima no eixo Y de 100 cm, menor que a máxima de 125 cm da gravação 11 na Tabela 4.2 da Seção 4.1.

Uma vez que o algoritmo de posicionamento é o mesmo do apresentado na Seção 3.8, nota-se comportamento e tendências similares do posicionamento estimado e erros associados a cada eixo. O eixo Y continua apresentando um viés de cálculo, possivelmente oriundo da calibração realizada, o que resulta em estimativas com maior erro. O eixo X novamente possui o segundo melhor resultado na estimação, enquanto que o eixo Z, apesar de possuir os menores resultados de erro, novamente apresenta uma tendência a ser calculado com alturas verticais acima da correta.

Ainda analisando os resultados das Tabelas 5.1, 5.2 e 5.3, não é percebida nenhuma relação com o **Modo** escolhido para o sistema tensorial e o desempenho do posicionamento. Ou seja, a princípio, independente dos parâmetros de codificação adotados, uma vez decodificada corretamente a mensagem com as dimensões da tela, o sistema S2CP funciona corretamente.

É importante destacar que, além das vantagens do esquema tensorial apresentadas no início deste Capítulo, os diferentes **Modos** apresentados nestes testes reais permitem o sistema se adaptar a diferentes condições de operação, como presença de ruído elevado e distâncias maiores, não testadas nesta dissertação. Por outro lado, o esquema tensorial é mais complexo, necessitando de um maior poder computacional tanto no transmissor quanto no receptor. Novos testes e estudos envolvendo o esquema tensorial no sistema S2CP é uma das perspectivas do Capítulo 7.

## Capítulo 6

### Conclusões

Este trabalho trouxe um novo sistema de posicionamento *indoor* baseado na tecnologia *Screen-to-Camera* (S2C), o qual foi então batizado de *Screen-to-Camera Positioning* (S2CP). O princípio do novo sistema é a utilização de uma única tela (e.g., televisão ou monitor de computadores) como ponto de referência de posicionamento para um dispositivo móvel imbuído de uma câmera (e.g., *smartphone*). A ideia é que a tela, através de protocolos S2C, seja capaz de enviar dados de cabeçalho (e.g., bits de sincronia, número de identificação, informações sobre dimensões da tela) e dados úteis de uso geral (e.g., voz digital), que serão então processados pelo receptor. Essa abordagem única abre perspectivas para o uso do S2CP em ambientes ricos em telas e câmeras, como residências, edifícios comerciais e ambientes urbanos, sendo talvez o maior potencial econômico a desnecessidade de se utilizar equipamentos extras para posicionamento.

O esquema S2CP foi adaptado para dois protocolos de comunicação distintos, com impactos não somente na parte de comunicações mas também no desenvolvimento de etapas de visão computacional pelo receptor. Um dos protocolos é o padronizado código de barras e o outro foi um protocolo tensorial de comunicação S2C desenvolvido recentemente dentro do próprio grupo de pesquisa.

Para o sistema de posicionamento usando códigos de barras, foi inicialmente desenvolvido um algoritmo de processamento de imagens baseado em morfologia (incluindo um detector de bordas de Canny) para identificação da tela a partir de uma sequência de vídeo contendo quadros portadores de mensagens codificadas no padrão BC-128 com as dimensões físicas da tela. Das 20 gravações realizadas com a câmera do *Smartphone* Poco F5 em um ambiente de baixa luminosidade e pouca interferência luminosa de outros

equipamentos eletrônicos, limitado a distâncias entre 40 cm e 150 cm, o algoritmo foi capaz de identificar corretamente o monitor e decodificar a mensagem transmitida em 17 gravações, demonstrando um funcionamento aceitável que valida a proposta do trabalho.

Para o sistema de posicionamento usando o esquema tensorial, foi utilizada uma abordagem baseada na variação temporal de pixels nas regiões de interesse, a fim de distinguir a tela que está transmitindo dos demais elementos na cena capturada. De 12 gravações, tal algoritmo foi capaz de identificar corretamente a tela transmissora nas simulações e gravações realizadas, assim como isolar e recortar corretamente a região exclusiva dos símbolos para correção de sua perspectiva e eventual decodificação dos dados em 11 gravações, validando o funcionamento da técnica proposta sob as condições de teste.

Independentemente do esquema de comunicação utilizado no sistema S2CP, para o cálculo das coordenadas da câmera em relação à tela, onde o centro da tela é a origem do sistema de coordenadas tridimensional, foi utilizada a matriz de transformação obtida na calibração da câmera, os pontos de imagem e os pontos de objeto para resolver o problema P4P, a partir das dimensões reais transmitidas da tela, tornando possível encontrar solução única para a estimação da pose da câmera em relação à tela. Através de experimentos com um notebook servindo de transmissor e um smartphone como receptor, os resultados mostraram que o sistema proposto pode alcançar resultados precisos para posicionamento 3D de curto alcance, com erros de posicionamento na ordem de 10 cm.

# Capítulo 7

## Perspectivas

A finalização deste trabalho pioneiro com o sistema S2CP deixou abertas diversas possibilidades para futuras pesquisas que possam expandir os códigos desenvolvidos e os resultados alcançados até aqui. Investigações futuras podem se concentrar no aprimoramento da robustez do sistema em ambientes dinâmicos, com um grande número de telas e sob condições de iluminação variadas. Além disso, o S2CP pode ser aprimorado com protocolos de comunicação mais robustos e de maior velocidade, ampliando ainda mais seu potencial para posicionamento em tempo real e transmissão de dados em cenários internos complexos.

### 7.1 Posicionamento com Múltiplas Telas

O esquema S2CP desenvolvido neste mestrado utilizou apenas uma tela para posicionamento e localização. Apesar dos resultados satisfatórios, especialmente para a validação da tecnologia proposta, a expansão do S2CP para o uso de múltiplas telas, seja de forma simultânea ou individualmente dentro de uma mesma rede de comunicação, agrega maior flexibilidade e amplia os cenários de aplicação em sistemas *IoT*.

Além dos quadros de sincronia, o protocolo tensorial pode transmitir quadros de identificação única para cada tela dentro de uma mesma rede *indoor*, permitindo que dispositivos móveis (e.g., *drones*) associem seus dados de posicionamento a diferentes equipamentos e realizem mapeamentos, por exemplo.

No caso de múltiplas telas transmitindo dados ao mesmo tempo (ou em intervalos curtos), o receptor pode aplicar técnicas de trilateração e cálculos de posicionamento médio para

aprimorar a precisão de sua localização. Além disso, caso cada tela transmita não apenas informações de posicionamento, mas também outros tipos de dados, a câmera poderá receber uma quantidade maior de informações, ampliando suas possibilidades de uso para cenários que necessitem maiores taxas de dados.

## 7.2 Flexibilização dos dados adicionais

Até o momento, somente dados de sincronização das mensagens e informações sobre as dimensões da tela foram transmitidos. Um dos principais potenciais de inovação do sistema S2CP está em sua capacidade de transmissão de dados de forma espacial e temporal, de maneira escalável com as dimensões dos equipamentos – ou seja, resolução da tela e do sensor de imagem. Tal potencial foi mencionado ao longo de todo o desenvolvimento e escrita deste trabalho e permanece como uma possibilidade para futuras pesquisas que explorem a transmissão de outros dados além de sincronização e posicionamento.

Um exemplo de dados complementares seria no monitoramento da higienização de ambientes em um hospital. O esquema S2CP com o protocolo tensorial poderia orientar de forma eficiente os *robôs* assistentes, indicando as áreas que demandam prioridade para a limpeza automatizada do piso.

## 7.3 Otimização do esquema S2CP com protocolo tensorial

Até a conclusão deste trabalho, a implementação do protocolo tensorial ainda se encontrava em estágios relativamente iniciais, deixando espaço para possíveis aprimoramentos e otimizações em sua estrutura e funcionamento, sobretudo porque ainda há uma evolução dos trabalhos propostos na própria melhoria do protocolo tensorial.

Para uma otimização do esquema S2CP com este protocolo, novos estudos podem se concentrar na avaliação dos diferentes fatores que podem determinar o desempenho do sistema, como:

- Distância e angulação entre transmissor e receptor, podendo gerar erros de perspectivas e aumentar a taxa de erro de bit do protocolo tensorial;
- Qualidade do sensor de imagem, incluindo sua taxa de captura de quadros, resolução e parâmetros intrínsecos;

- Qualidade do transmissor de imagem (tela), considerando sua resolução, tipo de tela (*LED, OLED, LCD IPS*) e a taxa de atualização de quadros;
- Desvios de sincronia entre as taxas de quadros da tela e da câmera.

Seguindo os princípios da teoria de sistemas de comunicação, onde a técnica de modulação é frequentemente ajustada para taxas de *bit* menores em condições adversas do canal, uma abordagem inovadora seria o desenvolvimento de um controle dinâmico da taxa de transmissão e das dimensões físicas dos tensores transmitidos. Esse ajuste poderia aprimorar a identificação e o processamento dos dados, além de contribuir para a redução da *Bit Error Rate* (BER).

## 7.4 Embarcar em Vídeo Real

Um dos propósitos da tecnologia S2C é a de transmitir dados de comunicação através dos quadros de um vídeo exibido em uma tela. Nesse sentido, uma das vantagens do protocolo tensorial em relação à transmissão por código de barras é sua capacidade de desempenhar uma função dupla: transmitir simultaneamente dados e imagens de forma, a princípio, dissociada. No caso dos códigos de barras, a informação visual percebida pelo olho humano está diretamente relacionada aos caracteres por ele transmitidos.

Portanto, considerando a aplicação efetiva em um sistema real, um trabalho com grande potencial de contribuição seria aplicar o S2CP com o protocolo tensorial em um vídeo real, para analisar o desempenho e a viabilidade do sistema sob duas perspectivas:

- **Avaliação objetiva:** Analisar a otimização e o desempenho do sistema considerando o erro de posicionamento e as taxas de erro de *bit* em vídeos reais, cujos pixels seguem distribuições estatísticas próprias
- **Avaliação subjetiva:** avaliar o impacto do sistema S2CP na percepção humana, uma vez que a função primária das telas é a de transmitir imagens e não informações de posicionamento.

## Referências bibliográficas

ARTH, C. et al. Exploiting Sensors on Mobile Phones for Augmented Reality. In: 2011 International Symposium on Mixed and Augmented Reality (ISMAR). [S.l.: s.n.], 2011. P. 195–204. DOI: 10.1109/ISMAR.2011.6092385.

ASHOK, A. et al. Capacity of screen-camera communications under perspective distortions. In: PB. PERVASIVE and Mobile Computing. [S.l.: s.n.], 2015. v. 16, p. 239–250. DOI: 10.1016/j.pmcj.2014.11.003. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1574119214001849>>.

AZUMA, R. T. A Survey of Augmented Reality. **Presence: Teleoperators and Virtual Environments**, v. 6, n. 4, p. 355–385, 1997. DOI: 10.1162/pres.1997.6.4.355.

BAI, L. et al. Camera Assisted Received Signal Strength Ratio Algorithm for Indoor Visible Light Positioning. **IEEE Communications Letters**, v. 23, n. 11, p. 2022–2025, nov. 2019. ISSN 1558-2558. DOI: 10.1109/LCOMM.2019.2935713. Disponível em: <<https://ieeexplore.ieee.org/document/8801944>>.

BOIXADER, A.; LABELLA, C.; CATALAN, M.; PARADELLS, J. A High-Accuracy, Scalable and Affordable Indoor Positioning System Using Visible Light Positioning for Automated Guided Vehicles. **Electronics**, v. 13, n. 1, 2024. ISSN 2079-9292. DOI: 10.3390/electronics13010082. Disponível em: <<https://www.mdpi.com/2079-9292/13/1/82>>.

BOUCHET, O. et al. Visible-light communication system enabling 73 Mb/s data streaming. In: p. 1042–1046. DOI: 10.1109/GLOCOMW.2010.5700092.

CANNY, J. A Computational Approach to Edge Detection. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, PAMI-8, n. 6, p. 679–698, nov. 1986. ISSN 1939-3539. DOI: 10.1109/TPAMI.1986.4767851.

CHEN, S. D.; RAMLI, A. R. Contrast enhancement using recursive mean-separate histogram equalization for scalable brightness preservation. **Consumer Electronics, IEEE Transactions on**, v. 49, p. 1301–1309, 2003. DOI: 10.1109/TCE.2003.1261233.

CODING DADDY. **CODE 128 CHARSET Table**. [S.l.: s.n.]. Accessed: 2025-01-22. Disponível em: <<https://www.coding-daddy.com/gs1/code128.html#code-128>>.

DAI, J. S. Euler–Rodrigues formula variations, quaternion conjugation and intrinsic connections. **Mechanism and Machine Theory**, v. 92, p. 144–152, 2015. ISSN 0094-114X. DOI: <https://doi.org/10.1016/j.mechmachtheory.2015.03.004>. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0094114X15000415>>.

DREWES, H.; PFEUFFER, K.; ALT, F. Time- and space-efficient eye tracker calibration. In: p. 1–8. DOI: 10.1145/3314111.3319818.

DURRANT-WHYTE, H.; BAILEY, T. Simultaneous Localization and Mapping: Part I. **IEEE Robotics & Automation Magazine**, v. 13, n. 2, p. 99–110, 2006. DOI: 10.1109/MRA.2006.1638022.

ELHOUSNI, M.; HUANG, X. A Survey on 3D LiDAR Localization for Autonomous Vehicles. **arXiv preprint arXiv:2007.00291**, 2020. Accepted by IV2020. arXiv: 2007.00291 [cs.CV].

EXPRESS CORPORATION. **Specification for Barcode Symbology**. [S.l.: s.n.]. Accessed: 2025-01-16. Disponível em: <<https://www.expresscorp.com/wp-content/uploads/2023/02/USS-128.pdf>>.

FAUGERAS, O. Three-dimensional computer vision: a geometric viewpoint. **MIT press**, jan. 1993.

FENG, M. et al. Design of OCC Indoor Positioning System Based on Flat Panel Light and Angle Sensor Assistance. **Applied Sciences**, v. 13, n. 8, p. 4745, abr. 2023. ISSN 2076-3417. DOI: 10.3390/app13084745.

GALLEGO, G.; DELBRÜCK, T. et al. Event-based Vision: A Survey. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 43, n. 6, p. 433–467, 2020. DOI: 10.1109/TPAMI.2020.3008413.

GALLEGO, G.; YEZZI, A. J. A compact formula for the derivative of a 3-D rotation in exponential coordinates. **Journal of Mathematical Imaging and Vision**, v. 51, p. 378–384, 2014.

GAO, X.-S.; HOU, X.-R.; TANG, J.; CHENG, H.-F. Complete solution classification for the perspective-three-point problem. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, IEEE, v. 25, n. 8, p. 930–943, 2003.

GONZALEZ, R. C.; WOODS, R. E. Processamento Digital de Imagens. In: 3. ed. São Paulo: Pearson, 2010. cap. 6.

GONZALEZ, R. C.; WOODS, R. E. Processamento Digital de Imagens. In: 3. ed. São Paulo: Pearson, 2010. cap. 9.

GONZALEZ, R. C.; WOODS, R. E. Processamento Digital de Imagens. In: 3. ed. São Paulo: Pearson, 2010. cap. 2.

GUAN, W. et al. Robot Localization and Navigation Using Visible Light Positioning and SLAM Fusion. **Journal of Lightwave Technology**, v. 39, n. 22, p. 7040–7051, nov. 2021. ISSN 15582213. DOI: 10.1109/JLT.2021.3113358.

HAAS, H.; SARBAZI, E.; MARSHOUD, H.; FAKIDIS, J. Visible-light communications and light fidelity. In: WILLNER, A. E. (Ed.). **Optical Fiber Telecommunications VII**. [S.l.]: Academic Press, 2019. P. 443–493. ISBN 9780128165027. DOI: 10.1016/B978-0-12-816502-7.00013-0. Disponível em: <<https://www.sciencedirect.com/science/article/pii/B9780128165027000130>>.

HADDAD, R. A.; AKANSU, A. N. A Class of Fast Gaussian Binomial Filters for Speech and Image Processing. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. 39, p. 723–727, mar. 1991.

HAMAGAMI, R. et al. Rolling-Shutter Sensor-Based Visible Light Communication with Cross-Screen Filter: Communication and Positioning System Using a Commercial Camera. In: 2021 IEEE 10th Global Conference on Consumer Electronics, GCCE 2021. [S.l.: s.n.], out. 2021. P. 386–390. ISBN 9781665436762. DOI: 10.1109/GCCE53005.2021.9621902.

HAO, J.; CHEN, J.; WANG, R. Visible Light Positioning Using A Single LED Luminaire. **IEEE Photonics Journal**, v. 11, n. 5, p. 1–13, 2019. DOI: 10.1109/JPHOT.2019.2930209.

HOLKER, M. Radiowave propagation. In: MAZDA, F. (Ed.). **Telecommunications Engineer's Reference Book**. [S.l.]: Butterworth-Heinemann, 1993. P. 27–1–27–15. ISBN 978-0-7506-1162-6. DOI: 10.1016/b978-0-7506-1162-6.50033-2. Disponível em: <<https://www.sciencedirect.com/science/article/pii/B9780750611626500332>>.

IEEE Approved Draft Standard for Local and Metropolitan Area Networks - Part 15.7: Short-Range Optical Wireless Communications Amendment 1: Higher Speed, Longer Range Optical Camera Communication (OCC). **IEEE P802.15.7a/D8, July 2024**, p. 1–49, 2024.

IEEE Draft Standard for Local and Metropolitan Area Networks - Part 15.7: Short-Range Optical Wireless Communications Amendment 1: Higher Speed, Longer Range Optical Camera Communication (OCC). **IEEE P802.15.7a/D8, July 2024**, p. 1–49, 2024.

IEEE STANDARD ASSOCIATION. IEEE Standard for Local and metropolitan area networks—Part 15.7: Short-Range Optical Wireless Communications (802.15.7-2018). **IEEE Std 802.15.7-2018 (Revision of IEEE Std 802.15.7-2011)**, p. 1–407, abr. 2011. DOI: 10.1109/IEEESTD.2019.8697198. Disponível em: <[https://standards.ieee.org/standard/802%7B%5C\\_%7D15%7B%5C\\_%7D7-2018.html](https://standards.ieee.org/standard/802%7B%5C_%7D15%7B%5C_%7D7-2018.html)>.

ITU-T. **Recommendation BT.601-7: Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios**. Geneva, Switzerland, 2011.

JIANG, W.; HAN, B.; HABIBI, M. A.; SCHOTTEN, H. D. **The road towards 6G: A comprehensive survey**. v. 2. [S.l.: s.n.], 2021. P. 334–366. DOI: 10.1109/OJCOMS.2021.3057679. arXiv: 2102.01420.

KANG, T.-W.; CHOI, Y.-S.; JUNG, J.-W. Estimation of Relative Position of Drone using Fixed Size QR Code. In: 2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC). Prague, Czech Republic: [s.n.], 2022. P. 442–447. DOI: 10.1109/SMC53654.2022.9945142.

KAPLAN, E. D.; HEGARTY, C. J. **Understanding GPS: Principles and Applications**. [S.l.]: Artech House, 2006.

KEYENCE CORPORATION. **CODE 128 e GS1-128**. [S.l.: s.n.]. Accessed: 2025-01-16. Disponível em: <[https://www.keyence.com.br/ss/products/auto\\_id/barcode\\_lecture/basic/code128/](https://www.keyence.com.br/ss/products/auto_id/barcode_lecture/basic/code128/)>.

KHURPADE, J. M.; RAO, D.; SANGHAVI, P. D. A Survey on IOT and 5G Network. In: 2018 International Conference on Smart City and Emerging Technology, ICSCET 2018. [S.l.]: IEEE, jan. 2018. P. 1–3. ISBN 9781538611852. DOI: 10.1109/ICSCET.2018.8537340.

KIM, J.-I.; GANG, H.-S.; PYUN, J.-Y.; KWON, G.-R. Implementation of QR Code Recognition Technology Using Smartphone Camera for Indoor Positioning. **Energies**, v. 14, n. 2759, 2021. DOI: 10.3390/en14102759. Disponível em: <<https://doi.org/10.3390/en14102759>>.

LI, Y. et al. A VLC Smartphone Camera Based Indoor Positioning System. **IEEE Photonics Technology Letters**, v. 30, n. 13, p. 1171–1174, jul. 2018. ISSN 1041-1135. DOI: 10.1109/LPT.2018.2834930. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-850467928357B5C&doi=10.1109%7B%5C%7D2FLPT.2018.2834930%7B%5C%7DpartnerID=40%7B%5C%7Dmd5=9fe2cf4094d44b8998e2bababb99f43%20https://ieeexplore.ieee.org/document/8357568/>>.

LIN, B. et al. An Indoor Visible Light Positioning System Based on Optical Camera Communications. **IEEE Photonics Technology Letters**, v. 29, n. 7, p. 579–582, 2017. DOI: 10.1109/LPT.2017.2669079. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-850157984717B5C&doi=10.1109%7B%5C%7D2FLPT.2017.2669079%7B%5C%7DpartnerID=40%7B%5C%7Dmd5=22936a207b465ac39691507ab98fcae0>>.

LIN, X.; ZHANG, L. Intelligent and Practical Deep Learning Aided Positioning Design for Visible Light Communication Receivers. **IEEE Communications Letters**, v. 24, n. 3, p. 577–580, mar. 2020. ISSN 15582558. DOI: 10.1109/LCOMM.2019.2958629.

LU, X. X. A Review of Solutions for Perspective-n-Point Problem in Camera Pose Estimation. **Journal of Physics: Conference Series**, IOP Publishing, v. 1087, p. 052009, 2018. DOI: 10.1088/1742-6596/1087/5/052009. Disponível em: <<https://doi.org/10.1088/1742-6596/1087/5/052009>>.

MAPUNDA, G. A. et al. Indoor Visible Light Communication: A Tutorial and Survey. **Wireless Communications and Mobile Computing**, v. 2020, n. 1, p. 8881305, 2020. DOI: <https://doi.org/10.1155/2020/8881305>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1155/2020/8881305>. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1155/2020/8881305>>.

MARCHAND, E.; UCHIYAMA, H.; SPINDLER, F. Pose Estimation for Augmented Reality: A Hands-On Survey. **IEEE Transactions on Visualization and Computer Graphics**, v. 22, n. 12, p. 2633–2651, 2016. DOI: 10.1109/TVCG.2015.2513408.

MARTÍNEZ-CIRO, R. A.; LÓPEZ-GIRALDO, F. E.; LUNA-RIVERA, J. M.; RAMÍREZ-AGUILERA, A. M. An Indoor Visible Light Positioning System for Multi-Cell Networks. **Photonics**, v. 9, n. 3, 2022. ISSN 2304-6732. DOI: 10.3390/photonics9030146. Disponível em: <<https://www.mdpi.com/2304-6732/9/3/146>>.

MASON, S. Computer vision and image processing. **Isprs Journal of Photogrammetry and Remote Sensing - ISPRS J PHOTOGRAMM**, v. 48, p. 24–25, 1993. DOI: 10.1016/0924-2716(93)90037-N.

- MEER, P.; GEORGESCU, B. Edge detection with embedded confidence. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 23, n. 12, p. 1351–1365, 2001. ISSN 1939-3539. DOI: 10.1109/34.977560.
- MENDOZA-SILVA, G. M.; TORRES-SOSPEDRA, J.; HUERTA, J. A meta-review of indoor positioning systems. **Sensors (Switzerland)**, v. 19, n. 20, 2019. DOI: 10.3390/s19204507.
- MERRY, K.; BETTINGER, P. Smartphone GPS accuracy study in an urban environment. **PLOS ONE**, v. 14, n. 7, e0219890, jul. 2019. ISSN 1932-6203. DOI: 10.1371/journal.pone.0219890. Disponível em: <<https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0219890>>.
- MONICO, J. F. G. **Posicionamento GNSS: descrição, fundamentos e aplicações**. [S.l.]: UNESP, 2008.
- NIE, S.; JORNET, J. M.; AKYILDIZ, I. F. Intelligent Environments Based on Ultra-massive Mimo Platforms for Wireless Communication in Millimeter Wave and Terahertz Bands. In: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). [S.l.: s.n.], mai. 2019. P. 7849–7853. DOI: 10.1109/ICASSP.2019.8683394.
- O'BRIEN, D. C. et al. Visible Light Communications: Challenges and possibilities. In: IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC. [S.l.]: IEEE, set. 2008. P. 1–5. ISBN 9781424426447. DOI: 10.1109/PIMRC.2008.4699964.
- OPENCV (Ed.). **Eroding and Dilating**. 2024. Disponível em: <[https://docs.opencv.org/3.4/db/df6/tutorial\\_erosion\\_dilatation.html](https://docs.opencv.org/3.4/db/df6/tutorial_erosion_dilatation.html)>. Acesso em: 11 dez. 2024.
- OPENCV (Ed.). **Finding contours in your image**. 2024. Disponível em: <[https://docs.opencv.org/3.4/df/d0d/tutorial\\_find\\_contours.html](https://docs.opencv.org/3.4/df/d0d/tutorial_find_contours.html)>. Acesso em: 11 dez. 2024.
- OPENCV (Ed.). **Image Thresholding**. 2024. Disponível em: <[https://docs.opencv.org/4.x/d7/d4d/tutorial\\_py\\_thresholding.html](https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html)>. Acesso em: 11 dez. 2024.
- OPENCV CONTRIBUTORS. **OpenCV Camera Calibration**. [S.l.: s.n.], n.d. [https://docs.opencv.org/4.x/dc/dbb/tutorial\\_py\\_calibration.html](https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html). Accessed: March 28, 2024.
- OPENCV CONTRIBUTORS. **OpenCV Morphological Transformations**. Accessed: January 15, 2025.
- ORG, P. (Ed.). **Pandas Array**. 2024. Disponível em: <<https://pandas.pydata.org/docs/reference/api/pandas.array.html>>. Acesso em: 11 dez. 2024.
- PAN, G. et al. 3-D Positioning System Based QR Code and Monocular Vision. In: 2020 5th International Conference on Robotics and Automation Engineering (ICRAE). Singapore, Singapore: [s.n.], 2020. P. 54–58. DOI: 10.1109/ICRAE50850.2020.9310908.
- PATHAK, P. H.; FENG, X.; HU, P.; MOHAPATRA, P. **Visible Light Communication, Networking, and Sensing: A Survey, Potential and Challenges**. v. 17. [S.l.: s.n.], mai. 2015. P. 2047–2077. DOI: 10.1109/COMST.2015.2476474.

PIXEL, S. a (Ed.). **The Perspective and Orthographic Projection Matrix**. Disponível em: <<https://www.scratchapixel.com/lessons/3d-basic-rendering/perspective-and-orthographic-projection-matrix/building-basic-perspective-projection-matrix.html>>. Acesso em: 28 mar. 2024.

QUAN, L.; LAN, Z. Linear N-point camera pose determination. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 21, n. 8, p. 774–780, 1999. DOI: 10.1109/34.784291.

QUINTANA, X. et al. Image sensors for digital photography: a short course for undergraduates. *I: Optics*. **Optica Pura y Aplicada**, v. 53, p. 1–23, mar. 2020. DOI: 10.7149/OPA.53.1.51040.

RAJ, A. et al. A Review–Unguided Optical Communications: Developments, Technology Evolution, and Challenges. **Electronics (Switzerland)**, v. 12, n. 8, 2023. DOI: 10.3390/electronics12081922.

RODRIGUES, I. S. C.; XIMENES, L. R. Space-time-color coding for QuadLED Color Shift Keying Modulation for Visible Light Communications. In: PROCEEDINGS - 2020 IEEE Latin-American Conference on Communications, LATINCOM 2020. [S.l.]: IEEE, nov. 2020. P. 1–6. ISBN 9781728189031. DOI: 10.1109/LATINCOM50620.2020.9282313.

ROTH, M. Roth, M.W.: Survey of neural network technology for automatic target recognition. *IEEE Transactions on Neural Networks* 1(1), 28-43. **Neural Networks, IEEE Transactions on**, v. 1, p. 28–43, 1990. DOI: 10.1109/72.80203.

SAEED, N. et al. Optical camera communications: Survey, use cases, challenges, and future trends. **Physical Communication**, v. 37, p. 100900, 2019. ISSN 18744907. DOI: 10.1016/j.phycom.2019.100900. arXiv: 1812.01259. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1874490719304859>>.

SEJAN, M. A. S. et al. A Comprehensive Survey on MIMO Visible Light Communication: Current Research, Machine Learning and Future Trends. **Sensors**, v. 23, n. 2, 2023. ISSN 1424-8220. DOI: 10.3390/s23020739. Disponível em: <<https://www.mdpi.com/1424-8220/23/2/739>>.

SESYUK, A.; IOANNOU, S.; RASPOPOULOS, M. A Survey of 3D Indoor Localization Systems and Technologies. **Sensors**, v. 22, n. 23, p. 9380, dez. 2022. ISSN 1424-8220. DOI: 10.3390/s22239380.

SHANMUGAVADIVU, P.; BALASUBRAMANIAN, K. Thresholded and Optimized Histogram Equalization for contrast enhancement of images. **Computers & Electrical Engineering**, v. 40, p. 757–768, 2014. DOI: 10.1016/j.compeleceng.2013.06.013.

SHAPIRO, L. G.; STOCKMAN, G. C. **Computer Vision**. [S.l.]: Prentice Hall, 2001. P. 137, 150.

SINGH, H. **How do Computers Store Images?** Analytics Vidhya. 2023. Disponível em: <<https://www.analyticsvidhya.com/blog/2021/03/grayscale-and-rgb-format-for-storing-images/>>. Acesso em: 24 abr. 2024.

SOBEL, I. An Isotropic 3x3 Image Gradient Operator. **Presentation at Stanford A.I. Project 1968**, fev. 2014.

STS. **Image Representation**. QuaXP. s/d. Disponível em: <<https://www3.tuhh.de/sts/hoou/data-quality-explored/2-1-1-image-representation.html>>. Acesso em: 24 abr. 2024.

SUNG, R.; JUNG, S.-h.; HAN, D. Sound based indoor and outdoor environment detection for seamless positioning handover. **ICT Express**, v. 1, n. 3, p. 106–109, dez. 2015. ISSN 24059595. DOI: 10.1016/j.ict.2016.02.001.

SUZUKI, S. et al. Topological structural analysis of digitized binary images by border following. **Computer Vision, Graphics, and Image Processing**, v. 30, n. 1, p. 32–46, 1985.

TEAM, O. **solvePnP() - OpenCV PNP Algorithm**. [S.l.], 2024. Accessed: 2024-10-09. Disponível em: <[https://docs.opencv.org/4.x/d9/d0c/group\\_\\_calib3d.html#ga549c2075fac14829ff4a58bc931c033d](https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html#ga549c2075fac14829ff4a58bc931c033d)>.

TELEDYNE PHOTOMETRICS. **How is an Image Made?** s/d. Disponível em: <<https://www.photometrics.com/learn/camera-basics/how-is-an-image-made>>. Acesso em: 4 mar. 2024.

TELI, S. R. et al. A SIMO hybrid visible-light communication system for optical IoT. **IEEE Internet of Things Journal**, v. 9, n. 5, p. 3548–3558, mar. 2022.

THRUN, S.; BURGARD, W.; FOX, D. **Probabilistic Robotics**. Cambridge, MA: MIT Press, 2005. Módulos II e III.

TRAN, T. K. et al. Demonstration of A Visible Light Receiver Using Rolling-Shutter Smartphone Camera. In: INTERNATIONAL Conference on Advanced Technologies for Communications. [S.l.: s.n.], 2018. 2018-Octob, p. 214–219. ISBN 9781538661130. DOI: 10.1109/ATC.2018.8587521. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85060788313%7B%5C%7Ddoi=10.1109%7B%5C%7D2FATC.2018.8587521%7B%5C%7DpartnerID=40%7B%5C%7Dmd5=8363c0b5d55be9186f4b5bed09a71621>>.

VARSHNEY, U. 4G wireless networks. **IT Professional**, v. 14, n. 5, p. 34–39, set. 2012. ISSN 15209202. DOI: 10.1109/MITP.2012.71.

VATANSEVER, Z.; LIAN, J.; BRANDT-PEARCE, M. Indoor Mapping Using the VLC Channel State Information. In: CONFERENCE Record - Asilomar Conference on Signals, Systems and Computers. [S.l.]: IEEE, out. 2019. 2018-Octob, p. 428–432. ISBN 9781538692189. DOI: 10.1109/ACSSC.2018.8645459.

VIEIRA, M. et al. Optimizing Indoor Airport Navigation with Advanced Visible Light Communication Systems. **Sensors**, v. 24, n. 16, 2024. ISSN 1424-8220. DOI: 10.3390/s24165445. Disponível em: <<https://www.mdpi.com/1424-8220/24/16/5445>>.

WING, M. G.; EKLUND, A.; KELLOGG, L. D. Consumer-Grade Global Positioning System (GPS) Accuracy and Reliability. **Journal of Forestry**, v. 103, n. 4, p. 169–173, jun. 2005. ISSN 0022-1201. DOI: 10.1093/jof/103.4.169. Disponível em: <<https://academic.oup.com/jof/article/103/4/169/4598618>>.

WONG, K.-Y.; MENDONCA, P.; CIPOLLA, R. Camera calibration from surfaces of revolution. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 25, n. 2, p. 147–161, 2003. DOI: 10.1109/TPAMI.2003.1177148.

XIE, E. et al. Over 10 Gbps VLC for Long-Distance Applications Using a GaN-Based Series-Biased Micro-LED Array. **IEEE Photonics Technology Letters**, v. 32, n. 9, p. 499–502, mai. 2020. ISSN 19410174. DOI: 10.1109/LPT.2020.2981827.

XIMENES, L. R.; ALVES, M. F. Tensor-Based Screen-to-Camera Communications. **IEEE Communications Letters**, v. 27, n. 10, p. 2787–2791, 2023. ISSN 1558-2558. DOI: 10.1109/LCOMM.2023.3305836.

XIMENES, L. R.; LAREDO, B. A.; ARTHUR, R. Integrated data detection and video restoration for optical camera communications. **Digital Signal Processing**, v. 141, p. 104192, 2023. ISSN 1051-2004. DOI: <https://doi.org/10.1016/j.dsp.2023.104192>. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1051200423002877>>.

XIMENES, L. R.; CARDOSO, P. G. S. Esquema de comunicação Screen-to-Camera (S2C) tensorial em blocos de vídeos segmentados espacialmente. In: PROCEEDINGS of the 42nd Brazilian Symposium on Telecommunications and Signal Processing (SBrT). Belém, PA: SBrT, 2024.

XIMENES, L. R.; GONÇALVES, P.; ALVES, M. F.; ARTHUR, R. Achieving super-resolution reconstruction in Optical Camera Communications. In: PROCEEDINGS of the 19th International Symposium on Wireless Communication Systems (ISWCS). [S.l.]: IEEE, 2024.

YOKAR, V. N.; LE-MINH, H.; GHASSEMLOOY, Z.; WOO, W. L. Performance evaluation technique for screen-to-camera-based optical camera communications. **IET Optoelectronics**, v. 17, n. 4, p. 184–193, 2023. DOI: <https://doi.org/10.1049/ote2.12102>. eprint: <https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/ote2.12102>. Disponível em: <<https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/ote2.12102>>.

ZHANG, H. et al. High-Precision Indoor Visible Light Positioning Using Modified Momentum Back Propagation Neural Network with Sparse Training Point. **Sensors**, v. 19, n. 10, 2019. ISSN 1424-8220. DOI: 10.3390/s19102324. Disponível em: <<https://www.mdpi.com/1424-8220/19/10/2324>>.

ZHANG, P. et al. Constraints and recent solutions of optical camera communication for practical applications. **Photonics**, v. 10, n. 6, p. 1–19, mai. 2023.

ZHANG, Z. et al. 6G Wireless Networks: Vision, Requirements, Architecture, and Key Technologies. **IEEE Vehicular Technology Magazine**, v. 14, n. 3, p. 28–41, 2019. DOI: 10.1109/MVT.2019.2921208.

ZHANG, Z. A Flexible New Technique for Camera Calibration. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 22, p. 1330–1334, 2000. Disponível em: <<https://www.microsoft.com/en-us/research/publication/a-flexible-new-technique-for-camera-calibration/>>.

---

ZHUANG, Y. et al. A Survey of Positioning Systems Using Visible LED Lights. **IEEE Communications Surveys & Tutorials**, v. 20, n. 3, p. 1963–1988, 2018. ISSN 1553-877X. DOI: 10 . 1109 / COMST . 2018 . 2806558. Disponível em: <<https://ieeexplore.ieee.org/document/8292854>>.

# Anexo - Tabela de caracteres do *Barcode*

## 128

Numerical value	CODE A	CODE B	CODE C	Bar pattern	Numerical value	CODE A	CODE B	CODE C	Bar pattern
0	SP	SP	00		54	V	V	54	
1	!	!	01		55	W	W	55	
2	"	"	02		56	X	X	56	
3	#	#	03		57	Y	Y	57	
4	\$	\$	04		58	Z	Z	58	
5	%	%	05		59	[	[	59	
6	&	&	06		60	\	\	60	
7	'	'	07		61	]	]	61	
8	(	(	08		62	^	^	62	
9	)	)	09		63	_	_	63	
10	*	*	10		64	NUL	"	64	
11	+	+	11		65	SOH	a	65	
12	,	,	12		66	STX	b	66	
13	-	-	13		67	ETX	c	67	
14	.	.	14		68	EOT	d	68	
15	/	/	15		69	ENQ	e	69	
16	0	0	16		70	ACK	f	70	
17	1	1	17		71	BEL	g	71	
18	2	2	18		72	BS	h	72	
19	3	3	19		73	HT	i	73	
20	4	4	20		74	LF	j	74	
21	5	5	21		75	VT	k	75	
22	6	6	22		76	FF	l	76	
23	7	7	23		77	CR	m	77	
24	8	8	24		78	SO	n	78	
25	9	9	25		79	SI	o	79	

Figura 7.1: Códigos de caracteres BC-128, parte I. Adaptado de (CODING DADDY, s.d.)

26	:	:	26		80	DLE	p	80	
27	;	;	27		81	DC1	q	81	
28	<	<	28		82	DC2	r	82	
29	=	=	29		83	DC3	s	83	
30	>	>	30		84	DC4	t	84	
31	?	?	31		85	NAK	u	85	
32	@	@	32		86	SYN	v	86	
33	A	A	33		87	ETB	w	87	
34	B	B	34		88	CAN	x	88	
35	C	C	35		89	EM	y	89	
36	D	D	36		90	SUB	z	90	
37	E	E	37		91	ESC	{	91	
38	F	F	38		92	FS	┘	92	
39	G	G	39		93	GS	}	93	
40	H	H	40		94	RS	~	94	
41	I	I	41		95	US	DEL	95	
42	J	J	42		96	FNC 3	FNC 3	96	
43	K	K	43		97	FNC 2	FNC 2	97	
44	L	L	44		98	SHIFT	SHIFT	98	
45	M	M	45		99	CODE C	CODE C	99	
46	N	N	46		100	CODE B	FNC 4	CODE B	
47	O	O	47		101	FNC 4	CODE A	CODE A	
48	P	P	48		102	FNC 1	FNC 1	FNC 1	
49	Q	Q	49		103	START (CODE A)			
50	R	R	50		104	START (CODE B)			
51	S	S	51		105	START (CODE C)			
52	T	T	52						
53	U	U	53			STOP			

Figura 7.2: Códigos de caracteres BC-128, parte II. Adaptado de (CODING DADDY, s.d.)

# Apêndice A

## Editor de Perspectivas

O Editor de Perspectiva foi um código desenvolvido ao longo deste trabalho para auxiliar em processos de ajustes de perspectivas em gravações e simulações, de forma manual. Ele permite carregar um vídeo (recomenda-se em formato .mp4 ou .avi, dependendo da preocupação com perdas de compressão no vídeo) em um determinado quadro desejado, e realizar operações de homografia sobre tal quadro, para posteriormente serem aplicadas ao vídeo completo.

Tal ferramenta permite:

- Gerar dados de posicionamento sintéticos para validação de sistemas de posicionamento, através da transformação da perspectiva da câmera;
- Corrigir distorções de perspectiva de forma manual, estimando os coeficientes da matriz de homografia necessários para tal correção;
- Pré visualizar a transformação em um quadro para validação visual;
- Salvar um novo arquivo de vídeo com a homografia aplicada.

Além de seu uso neste trabalho, principalmente em ajustes de perspectivas nas Seções 3.3 e 5.2, este *software* pode ser usado em futuros trabalhos que envolvam processamento de imagens ou visão computacional, devido à sua facilidade de uso e versatilidade.

O sistema desenvolvido utiliza uma matriz de homografia (conforme aprofundado na Seção 2.9) de dimensões  $3 \times 3$  para realizar a transformação de perspectiva em quadros individuais do vídeo. Esta matriz representa uma transformação linear que mapeia pontos de uma imagem original para novos pontos em uma imagem transformada. O processo envolve:

- O carregamento de um quadro do vídeo, que é usado como referência para a aplicação das transformações.
- O uso de uma função para aplicar a transformação de perspectiva, corrigindo distorções geométricas.
- Preencher áreas vazias do quadro transformado com uma cor de fundo uniforme escura.

Para facilitar o alinhamento visual, o sistema permite desenhar uma grade sobre a imagem transformada conforme a Figura A.1. Essa grade auxilia visualmente para aplicar alterações e ajustar a transformação com maior precisão. A grade é parametrizada por tamanho das células, cor e espessura das linhas.



Figura A.1: Quadro principal do editor de vídeo. Um quadro do vídeo é exibido e sobreposto pela grade.

A *interface* gráfica foi desenvolvida utilizando a biblioteca Tkinter, com suporte a diferentes modos de edição:

- **Modo Deslizante (Sliders):** Permite ajustar os coeficientes da matriz de transformação diretamente por meio de controles deslizantes. Figura A.2.
- **Modo Matriz Visual:** Oferece campos de entrada para modificar manualmente os valores da matriz. Figura A.3.
- **Modo de Homografia Assistida:** Permite selecionar pontos de referência na imagem original e definir posições correspondentes na imagem transformada, calculando

automaticamente a matriz de homografia. Os pontos de origem e de destino devem ser selecionados na mesma ordem: superior esquerdo, superior direito, inferior esquerdo e inferior direito Figura A.4.

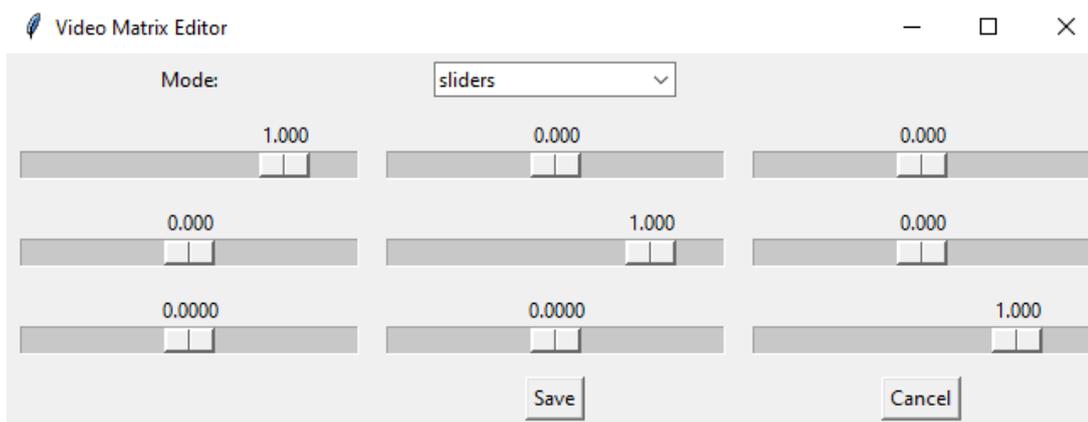


Figura A.2: Editor Modo Deslizante. Cada barra corresponde àquele elemento da matriz de homografia.

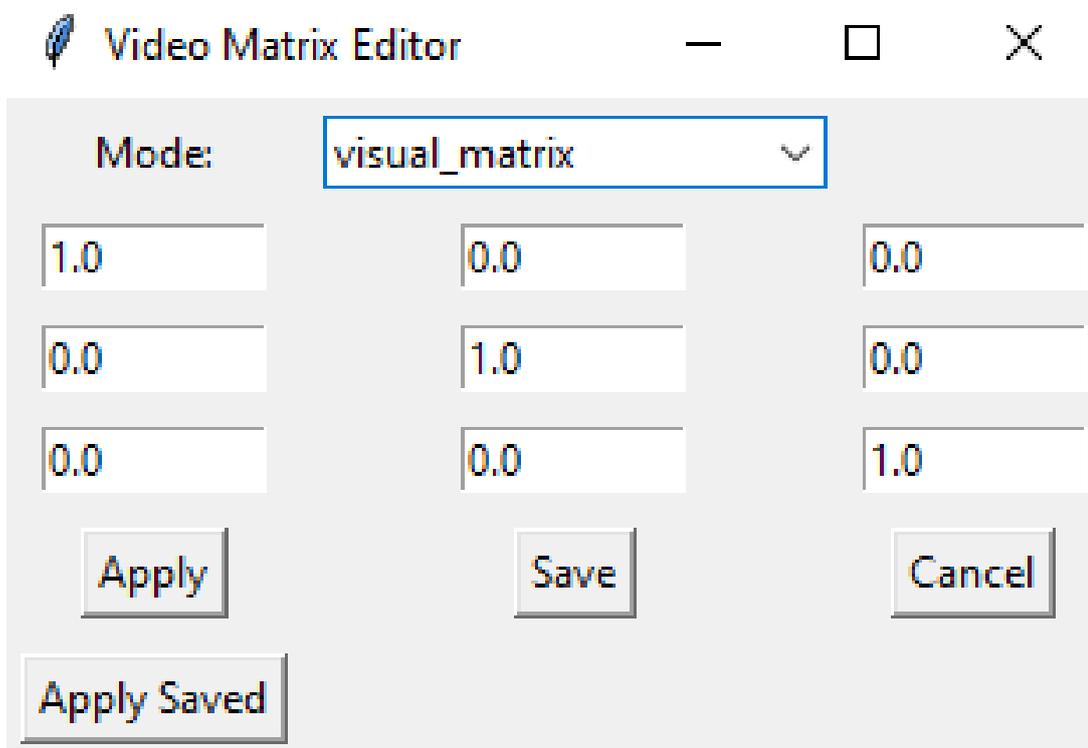


Figura A.3: Editor Modo Matriz Visual. Cada célula da matriz de homografia pode ser preenchida individualmente.

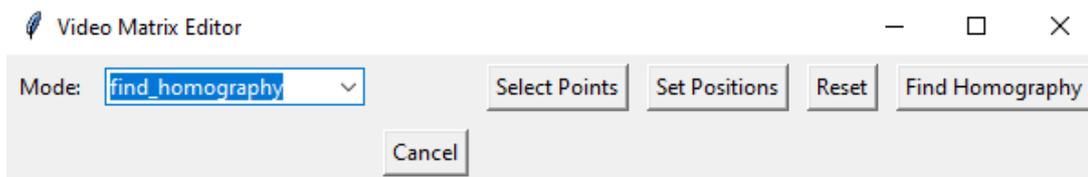


Figura A.4: Editor Modo de Homografia Assistida. Os pontos de origem e de destino devem ser selecionados na mesma ordem: superior esquerdo, superior direito, inferior esquerdo e inferior direito.

No modo de Homografia Assistida, o sistema suporta a seleção interativa de pontos de controle no quadro de referência e a definição de suas novas posições. A partir dessas correspondências, calcula-se a matriz de homografia utilizando algoritmos embutidos da função *findHomography* do OpenCV. A matriz calculada pode ser salva e aplicada posteriormente no modo Matriz Visual, com o botão *Apply Saved* da Figura A.3. Caso o resultado não seja satisfatório, pode-se apagar os pontos aplicados e refazer os ajustes manuais.

O funcionamento deste modo assistido pode ser observado na Figura A.5. Em **A**, foram selecionados os quatro pontos originais como referência ao clicar em "Select Points". Em **B**, foram selecionadas as novas coordenadas que deseja-se posicionar os pontos originais ao clicar em "Set Positions". Em **C**, a matriz de homografia estimada ao clicar em "Find Homography" é aplicada no Modo Matriz Visual com o botão *Apply Saved*. Em **D**, observa-se o quadro resultante com a nova perspectiva ao clicar em "Apply"

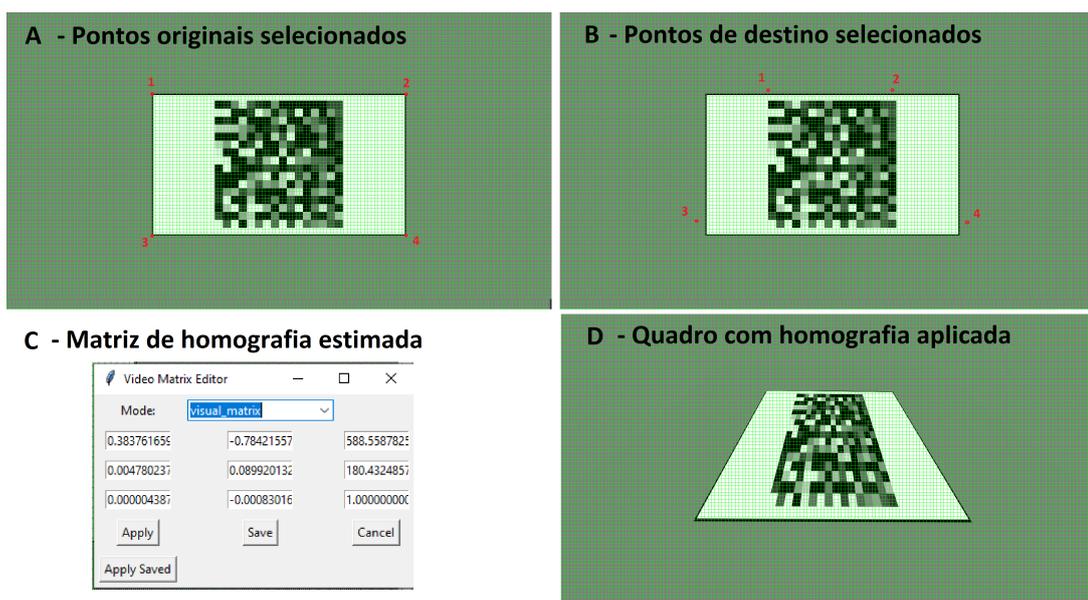


Figura A.5: Exemplo do Modo de Homografia Assistida.

O sistema permite a pré-visualização das transformações aplicadas a um quadro do vídeo, exibindo a imagem transformada com a grade sobreposta conforme a Figura A.6 Após ajustes satisfatórios, o vídeo transformado é salvo, aplicando a matriz final a todos os quadros do vídeo original. Também é possível apenas cancelar e fechar a ferramenta a qualquer momento.

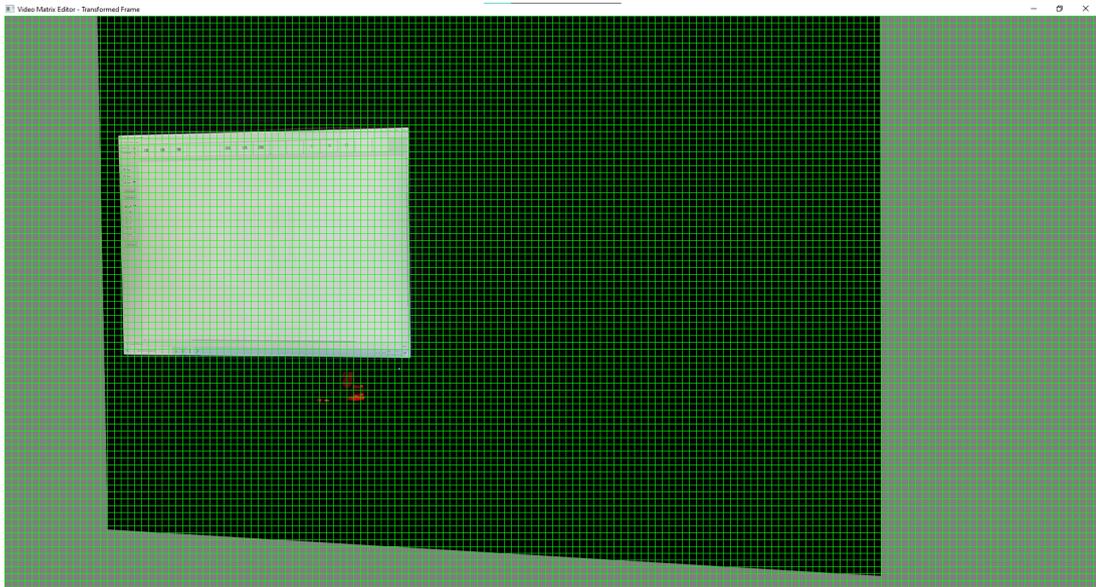


Figura A.6: Visualização do quadro transformado do vídeo após aplicar a homografia.



# Apêndice B

## Plano de Gestão de Dados

### Indoor drone positioning system using Screen-to-Camera and Visible Light Communication

#### Descrição dos Dados e Metadados

---

##### Quais serão os dados coletados?

Todos os dados gerados a partir desta pesquisa serão em formatos digitais e necessitarão de diferentes softwares e sistemas para acesso, importação e processamento.

Para o estudo de padrões e propriedades dos pixels exibidos em telas, serão coletadas imagens e vídeos através de câmeras de smartphones comercialmente definidas. Tais capturas conterão a tela como foco e serão feitas em diferentes distâncias e ângulos para análise de perspectivas variadas, medidas seguindo a convenção do Sistema Internacional de Unidades.

Os dados gerados a partir dessas coletas serão processados através do software Matlab em versão igual ou superior a R2017a e códigos de autoria dos pesquisadores em Python 3.9 e C++

#### Aspectos Legais e Facilidade de Acesso aos Dados

---

##### Quais são as questões legais e éticas associadas aos dados e relevantes a este projeto?

Por tratar-se de de uma pesquisa de âmbito tecnológico utilizando equipamentos disponibilizados comercialmente, testes dentro de ambientes exclusivamente dedicados aos estudos e simulações utilizando softwares acessíveis diretamente pelos pesquisadores, não há testes e estudos envolvendo seres vivos ou que apresentem riscos ao espaço utilizado, excluindo-se a necessidade de avaliação do Comitê de Ética.

Ao mesmo tempo, trata-se de uma pesquisa que vislumbra a inovação tecnológica, portanto a criação de uma Patente a partir dos resultados obtidos é uma possibilidade e neste caso deverá ser tratada com o devido protocolo de registro de patentes.

##### Quais são as políticas a serem utilizadas para o compartilhamento de dados?

Os dados gerados por esta pesquisa serão identificados de maneira persistente através de DOI. Tais dados serão compartilhados publicamente sem solicitações para visualização, porém partes interessadas em utilizá-los e/ou gerar novos dados manipulando os disponibilizados deverão solicitar ao pesquisador responsável pela coleta dos dados.

A pesquisa que gerou os dados deverá ser citada como referência, atribuindo o crédito aos pesquisadores envolvidos.

### **Gestão de Dados e Armazenamento**

---

**Em que formatos serão armazenados os arquivos resultantes da pesquisa em questão? Que software poderá ser utilizado para a manipulação de cada um dos formatos listados?**

Imagens serão coletadas no formato PNG e poderão ser importadas em softwares como Photoshop, GIMP e Paint. Vídeos serão armazenados no formato MP4 e poderão ser reproduzidos através de VLC, Windows Media Player, QuickTime e similares.

Conteúdos textuais serão disponibilizados no formato PDF e CSV para fácil leitura e importação através de ferramentas como Notepad, Vi e Emacs.

Códigos desenvolvidos através do Matlab em versão igual ou superior a R2017a necessitarão da utilização de tal software em versões compatíveis para importação. Programas desenvolvidos em Python 3.9 e C++ necessitarão de IDEs e/ou interpretadores adequados a essas linguagens de programação.

**Como e onde estes arquivos serão mantidos? Por quanto tempo ocorrerá esta preservação? Como será realizado o backup destes dados?**

Os dados serão mantidos acessíveis por no mínimo 2 anos após término da pesquisa dentro dos repositórios do Zenodo. Demais backups serão realizados mensalmente dentro do Google Drive e Github privados dos pesquisadores.

---