

Universidade Federal de Santa Catarina  
Centro Tecnológico, de Ciências Exatas e Educação  
Departamento de Engenharia de  
Controle e Automação e Computação



Hugo Ferrari Pegoretti

Rastreamento de Múltiplos Objetos em Vídeo Utilizando Filtro  
de Kalman

Blumenau

2021

**Hugo Ferrari Pegoretti**

**Rastreamento de Múltiplos Objetos em Vídeo  
Utilizando Filtro de Kalman**

Trabalho de Conclusão de Curso apresentado à Universidade Federal de Santa Catarina como parte dos requisitos necessários para a obtenção do Título de Engenheiro de Controle e Automação.  
Orientador: Prof. Dr. Marcos Vinicius Matsuo

Universidade Federal de Santa Catarina  
Centro Tecnológico, de Ciências Exatas e Educação  
Departamento de Engenharia de  
Controle e Automação e Computação

Blumenau  
2021

Hugo Ferrari Pegoretti

# Rastreamento de Múltiplos Objetos em Vídeo Utilizando Filtro de Kalman

Trabalho de Conclusão de Curso apresentado à Universidade Federal de Santa Catarina como requisito parcial para a obtenção do título de Engenheiro de Controle e Automação.

**Comissão Examinadora**

---

Prof. Dr. Marcos Vinicius Matsuo  
Universidade Federal de Santa Catarina  
Orientador

---

Prof. Dr. Leonardo Mejia Rincon  
Universidade Federal de Santa Catarina

---

Prof. Dr. Adão Boava  
Universidade Federal de Santa Catarina

Blumenau, 1 de outubro de 2021

Dedico este trabalho a minha mãe, Cleonice, meu pai, Valdir, minha irmã, Renata, e meu avô, Romeu.

# Agradecimentos

Agradeço aos meus pais, Cleonice e Valdir, e a minha irmã Renata, pelos valores que me ensinaram, a educação que me proporcionaram, apoio em minhas decisões e suporte nos momentos difíceis.

Agradeço aos meus amigos da vida e colegas da Universidade, que me ajudaram muito durante o período de graduação e deixaram os dias mais leves com os momentos alegres compartilhados. Aproveito para agradecer especialmente meu amigo Anthony, com o qual dividi inúmeros momentos durante a graduação e se tornou uma grande amizade para a vida.

Agradeço aos professores que fizeram parte da minha formação acadêmica e que foram fundamentais para que eu chegasse onde estou hoje. Em especial, agradeço ao meu orientador Marcos Vinicius Matsuo, por me auxiliar no desenvolvimento deste projeto e por compartilhar seu vasto conhecimento.

*"Se você acha que pode, ou que não pode fazer alguma coisa, você tem sempre razão."*  
(Henry Ford)

# Resumo

O rastreamento de múltiplos objetos em vídeo é um campo de estudo da área de visão computacional. Basicamente, um dos métodos de rastreamento de objetos mais utilizado consiste na execução de duas etapas: detecção dos objetos de interesse presentes em uma cena e estimação da posição destes objetos. Assim, este trabalho propõe a implementação de um sistema de visão computacional capaz de detectar e rastrear múltiplos objetos em um vídeo utilizando filtro de Kalman, sendo projetado para estimar de forma precisa a posição de um objeto de interesse, mesmo em situações em que a posição do objeto não pode ser medida (por sensores) em um dado instante de tempo. O sistema foi desenvolvido através do software Matlab, apresentando resultados satisfatórios para diferentes cenários de operação.

**Palavras-Chave:** 1. Detecção. 2. Rastreamento. 3. Filtro de Kalman.

# Abstract

Tracking multiple objects in video is an area of computer vision. One of the most used object tracking methods is the execution of two steps: detection of objects of interest present in a scene and estimation of the position of these objects. Thus, this work proposes the implementation of a computer vision system capable of detecting and tracking multiple objects in a video using a Kalman filter, being designed to accurately estimate the position of an object of interest, even in situations where the position object cannot be measured (by sensors) at a given instant of time. The system was developed using Matlab software, presenting satisfactory results for different operating scenarios.

**Keywords:** 1. Detection. 2. Tracking. 3. Kalman Filter.



# Lista de figuras

Figura 1 – Diferença entre as imagens digitais vetorial e raster. . . . .	16
Figura 2 – Imagem binária e em escala de cinza, respectivamente. . . . .	17
Figura 3 – Valores dos pixels das camadas RGB em uma região predominantemente verde. . . . .	17
Figura 4 – Decomposição da imagem original nas camadas R, G e B. . . . .	18
Figura 5 – Plano <i>ab</i> do espaço de cor CIELAB. . . . .	19
Figura 6 – Decomposição da imagem original nas camadas L, A e B. . . . .	20
Figura 7 – Etapas de processamento do algoritmo de segmentação de imagem baseado em cor. . . . .	21
Figura 8 – Exemplo de resultado obtido a partir da segmentação da imagem, tendo o vermelho como cor de referência. . . . .	22
Figura 9 – Exemplos de elementos estruturantes. . . . .	22
Figura 10 – Exemplo de resultado obtido por meio de operações morfológicas. . . .	23
Figura 11 – Exemplo de operação morfológica de erosão. . . . .	24
Figura 12 – Operações morfológicas: (a) Imagem de Entrada; (b) Dilatação; (c) Erosão; (d) Abertura; (e) Fechamento. . . . .	25
Figura 13 – Resultado da análise de componentes conectados. . . . .	25
Figura 14 – Análise de componentes conectados. (a) Imagem de entrada; (b) Componentes conectados. . . . .	26
Figura 15 – Componentes conectados apresentados separadamente. (a) Objeto 1; (b) Objeto 2; (c) Objeto 3. . . . .	26
Figura 16 – Exemplo de <i>bounding box</i> em torno de uma região de interesse. . . . .	27
Figura 17 – Centroide do objeto de interesse. . . . .	28
Figura 18 – Objeto se movimentando no plano $(u, v)$ . . . . .	29
Figura 19 – Fluxograma das etapas de processamento realizadas no sistema de rastreamento desenvolvido. . . . .	33
Figura 20 – Diagrama do processo de detecção e localização de objetos. . . . .	34
Figura 21 – Fluxograma com atribuição das posições medidas e gerenciamento dos filtros de Kalman. . . . .	36
Figura 22 – Quadros do vídeo utilizado no Exemplo 4. (a) tempo = 3.16 s; (b) tempo = 6.75 s; (c) tempo = 10.37 s; (d) tempo = 16.12 s. . . . .	38
Figura 23 – Quadros do vídeo do Exemplo 1. (a) tempo = 0.25 s; (b) tempo = 8.12 s; (c) tempo = 20.79 s. . . . .	39
Figura 24 – Quadros do vídeo 1 com rastreamento. (a) tempo = 1.41 s; (b) tempo = 5.99 s; (c) tempo = 20.79 s. . . . .	39

Figura 25 – Quadros do vídeo do Exemplo 2. (a) tempo = 3.12 s; (b) tempo = 5.99 s; (c) tempo = 19.25 s; (d) tempo = 20.79 s. . . . .	40
Figura 26 – Quadros do vídeo 2 com rastreamento. (a) tempo = 3.79 s; (b) tempo = 12.79 s; (c) tempo = 19.45 s; (d) tempo = 20.79 s. . . . .	41
Figura 27 – Quadros do vídeo do Exemplo 3. (a) tempo = 4.58 s; (b) tempo = 7.66 s; (c) tempo = 10.79 s; (d) tempo = 16.29 s . . . . .	41
Figura 28 – Quadros do vídeo 3 com rastreamento. (a) tempo = 8.20 s; (b) tempo = 10.25 s; (c) tempo = 13.41 s; (d) tempo = 17.75 s. . . . .	42
Figura 29 – Quadros do vídeo do Exemplo 4. (a) tempo = 6.04 s; (b) tempo = 9.66 s; (c) tempo = 15.58 s; (d) tempo = 16.54 s. . . . .	43
Figura 30 – Quadros do vídeo 4 com rastreamento. (a) tempo = 5.83 s; (b) tempo = 7.16 s; (c) tempo = 12.12 s; (d) tempo = 16.91 s. . . . .	43
Figura 31 – Quadros do vídeo do Exemplo 5. (a) tempo = 6.20 s; (b) tempo = 10.25 s; (c) tempo = 12.37 s; (d) tempo = 16.95 s. . . . .	44
Figura 32 – Quadros do vídeo 5 com rastreamento. (a) tempo = 8.37 s; (b) tempo = 11.12 s; (c) tempo = 13.25 s; (d) tempo = 17.79 s. . . . .	45

# Lista de tabelas

Tabela 1 – Procedimentos adotados para situações em que a quantidade de posições medidas é diferente do número de filtros de Kalman existentes. . .	37
---	----

# Lista de Siglas e Abreviaturas

RGB      *Red, Green e Blue*

# Lista de Símbolos

$u_k$	<i>Posição do objeto no eixo horizontal no instante <math>k</math>.</i>
$\dot{u}_k$	<i>Velocidade do objeto no eixo horizontal no instante <math>k</math>.</i>
$v_k$	<i>Posição do objeto no eixo vertical no instante <math>k</math>.</i>
$\dot{v}_k$	<i>Velocidade do objeto no eixo vertical no instante <math>k</math>.</i>
$\mathbf{x}_k$	<i>Vetor de estados no instante <math>k</math>.</i>
$\tilde{\mathbf{x}}_k$	<i>Estado predito no instante <math>k</math>.</i>
$\hat{\mathbf{x}}_k$	<i>Estado estimado no instante <math>k</math>.</i>
$\Delta T$	<i>Período de amostragem.</i>
$\mathbf{z}_k$	<i>Vetor de posições medidas no instante <math>k</math>.</i>
$\hat{\mathbf{z}}_k$	<i>Vetor de posições estimadas no instante <math>k</math>.</i>
$\tilde{\mathbf{P}}_k$	<i>Matriz de covariância do estado predito no instante <math>k</math>.</i>
$\hat{\mathbf{P}}_k$	<i>Matriz de covariância do estado estimado no instante <math>k</math>.</i>
$\mathbf{F}$	<i>Matriz de estados.</i>
$\mathbf{H}$	<i>Matriz de saída.</i>
$\mathbf{s}_k$	<i>Vetor de perturbação do vetor de estado no instante <math>k</math>.</i>
$\mathbf{w}_k$	<i>Vetor de perturbação da saída do sistema no instante <math>k</math>.</i>
$\hat{\mathbf{S}}$	<i>Matriz de covariância do ruído no modelo dinâmico.</i>
$\hat{\mathbf{W}}$	<i>Matriz de covariância do ruído nas posições medidas.</i>
$\mathbf{I}$	<i>Matriz identidade.</i>
$\mathbf{e}_k$	<i>Erro entre posição medida e posição predita no instante <math>k</math>.</i>
$\mathbf{K}_k$	<i>Ganho de Kalman no instante <math>k</math>.</i>

# Sumário

1	INTRODUÇÃO . . . . .	14
1.1	Objetivo geral . . . . .	14
1.2	Objetivos específicos . . . . .	14
1.3	Organização do trabalho . . . . .	15
2	FUNDAMENTAÇÃO TEÓRICA . . . . .	16
2.1	Imagem digital . . . . .	16
2.2	Espaços de cores RGB e CIELAB . . . . .	16
2.3	Processamento digital de imagens . . . . .	19
2.3.1	Segmentação de objetos baseado em cor . . . . .	20
2.3.2	Operações morfológicas . . . . .	21
2.3.3	Análise de componentes conectados . . . . .	24
2.3.4	Extração de características de região . . . . .	25
2.4	Filtro de Kalman . . . . .	28
3	DESENVOLVIMENTO . . . . .	33
3.1	Sistema proposto . . . . .	33
3.1.1	Detecção e localização de objetos . . . . .	34
3.1.2	Algoritmo de atribuição das posições e gerenciamento dos filtros de Kalman . . . . .	35
4	RESULTADOS . . . . .	38
4.1	Exemplo 1 . . . . .	38
4.2	Exemplo 2 . . . . .	40
4.3	Exemplo 3 . . . . .	41
4.4	Exemplo 4 . . . . .	42
4.5	Exemplo 5 . . . . .	44
4.6	Comentários gerais sobre a complexidade computacional do sistema implementado . . . . .	45
5	CONCLUSÕES . . . . .	47
5.1	Trabalhos futuros . . . . .	47
	REFERÊNCIAS BIBLIOGRÁFICAS . . . . .	48

# 1 Introdução

Em diversas aplicações é necessário realizar o rastreamento de objetos presentes em um vídeo, sendo tais aplicações encontradas em sistemas militares, soluções de engenharia para ambientes industriais e, também, no auxílio na segurança de pessoas e locais [1].

Para rastrear um objeto é necessário identificá-lo previamente no contexto em que está localizado. Desta forma, a etapa de detecção é essencial para o rastreamento adequado de um objeto de interesse, sendo esta etapa (em geral) a mais complexa e com maior custo computacional em comparação ao rastreamento, pois é nela em que é realizado o processo de distinção dos objetos presentes em uma cena. Em cenários de operações reais há um grande número de variáveis que podem causar interferências dificultando a detecção de um objeto, como a iluminação, por exemplo [2]. Por outro lado, a etapa de rastreamento, em geral, possui custo computacional menor por receber uma quantidade de informações de entrada inferior à etapa anterior.

Em particular, o rastreamento de objetos pode ser realizado por meio do filtro de Kalman, que trata-se de um estimador de estados capaz de determinar com precisão a posição de um objeto de interesse através de algumas etapas de processamento [3]. O filtro de Kalman é baseado em duas informações para seu funcionamento ideal: a posição do objeto de interesse obtida na etapa de detecção e o modelo dinâmico do sistema que descreve o comportamento do objeto [4, 5]. Adicionalmente, o filtro permite ponderar a estimativa da posição do objeto quando há incertezas em uma das duas informações citadas anteriormente. Desta forma, caso a detecção possua ruído elevado (comprometendo a medição da posição do objeto), é possível atribuir mais peso ao modelo dinâmico do sistema para obter uma estimativa de posição mais precisa. Sendo o contrário, também possível.

## 1.1 Objetivo geral

Este trabalho tem como objetivo geral implementar um sistema de rastreamento de objetos em vídeo, capaz de rastrear simultaneamente as posições de múltiplos objetos em cenários dinâmicos, isto é, com objetos entrando e saindo de cena.

## 1.2 Objetivos específicos

Os objetivos específicos deste trabalho são listados a seguir:

1. Compreender os conceitos básicos de imagem digital e espaço de cor.

2. Estudar técnicas e implementar algoritmos de processamento digital de imagens.
3. Estudar o filtro de Kalman.
4. Desenvolver um algoritmo de detecção e localização de objetos baseado em cor.
5. Desenvolver um algoritmo para rastreamento de objetos.
6. Avaliar os resultados obtidos.

### 1.3 Organização do trabalho

Este trabalho está organizado como segue. No Capítulo 2 são apresentados alguns conceitos relacionados com a fundamentação teórica deste trabalho, sendo abordadas a definição de imagem digital, peculiaridades e diferenças entre os espaços de cores RGB e CIELAB, técnicas de processamento digital de imagens, além do embasamento teórico sobre filtro de Kalman. No Capítulo 3 é apresentado o sistema de rastreamento de múltiplos objetos implementado. No Capítulo 4 são apresentados alguns resultados da aplicação do sistema proposto em diferentes cenários de operação, com a finalidade de avaliar seu desempenho. Por fim, no Capítulo 5 são elencadas as considerações finais e sugestões de trabalhos futuros.



## 2 Fundamentação teórica

### 2.1 Imagem digital

Uma imagem digital é a representação de uma imagem bidimensional por meio de um conjunto de valores discretos [6]. As imagens digitais são categorizadas de duas maneiras: vetorial e raster, sendo a primeira composta por elementos geométricos, como curvas, pontos, vértices e polígonos [7, 8]. Esse tipo de imagem digital é capaz de reorganizar seus elementos em situações de variação de escala, sem perder a qualidade original. Já a segunda é uma representação matricial, onde cada elemento da matriz é denominado como pixel. Pixels representam as menores unidades de uma imagem digital do tipo raster, sendo esta incapaz de preservar a qualidade original da imagem ao ser visualizada mais detalhadamente. Na Figura 1 são apresentados exemplos de imagens digitais dos tipos raster e vetorial, sendo visíveis as particularidades de cada tipo nas divisórias das regiões A, B e C.

Por possuir valores estáticos e facilmente mapeáveis (propriedades de cada pixel), as imagens do tipo raster são mais utilizadas no contexto de processamento digital de imagem. Desta forma, os valores dos pixels da imagem original podem ser utilizados para identificar objetos na cena analisada, ou ainda realizar modificações na coloração de algum objeto de interesse, por exemplo.

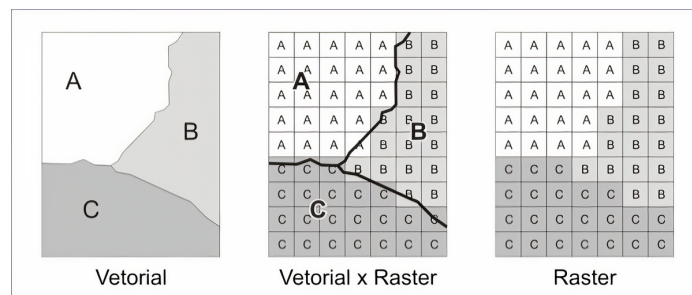


Figura 1 – Diferença entre as imagens digitais vetorial e raster.

Fonte – [9]

### 2.2 Espaços de cores RGB e CIELAB

Em particular, uma imagem digital raster pode ser classificada em relação a sua coloração como: binária, escala de cinza ou colorida. Na primeira categoria, os pixels da imagem podem possuir apenas os valores zero ou um, representando preto e branco, respectivamente. Por outro lado, em uma imagem em escala de cinza os pixels podem possuir

valores dentro de um dado intervalo, representando uma variação gradual entre o preto e o branco [10] (exemplos de imagens binária e em escala de cinza são apresentadas na Figura 2).



Figura 2 – Imagem binária e em escala de cinza, respectivamente.

Fonte – [11]

Diferentemente das imagens binárias e em escala de cinza (compostas por apenas uma camada de profundidade), as imagens coloridas são compostas por três camadas (matrizes), onde os valores dos pixels de cada camada representam a variação de algum parâmetro (dependendo do espaço de cor utilizado). Por exemplo, no espaço de cor RGB os valores armazenados em cada camada representam o nível de intensidade das cores vermelho (*red*), verde (*green*) e azul (*blue*), respectivamente [12].

Na Figura 3 é apresentada uma imagem colorida onde são destacados os valores dos pixels em uma dada região. Nesta imagem, nota-se que todos os pixels são compostos por um conjunto de valores nas três camadas citadas anteriormente: R, G e B. Na região destacada, percebe-se um valor elevado na camada correspondente a cor verde e valores inferiores nas demais camadas.

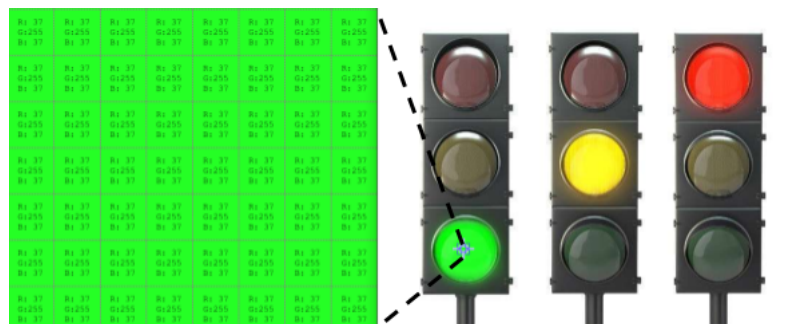


Figura 3 – Valores dos pixels das camadas RGB em uma região predominantemente verde.

Fonte – Fonte autoral

Ao analisar separadamente cada camada, observa-se alguns detalhes importantes para a percepção da composição final da imagem. Na Figura 4, são apresentadas as três camadas em escala de cinza da imagem anterior, onde as regiões mais claras correspondem a

valores maiores. Desta forma, a região em verde da imagem colorida original possui maior expressividade na camada G, sendo menos visível nas outras camadas (como esperado, de acordo com os valores apresentados na Figura 3). Também fica nítido que a região em vermelho da imagem colorida original é representada na camada R por meio de valores elevados. A cor amarela por ser uma combinação das cores vermelho e verde é expressa com notoriedade nas camadas R e G. Da mesma forma, a cor azul não possui destaque em nenhuma camada por não estar presente na cena. Já o fundo branco é expresso através de valores elevados nas três camadas, em contrapartida, observa-se que as regiões escuras (carcaça do semáforo) possuem valores próximos de zero em todas as camadas.

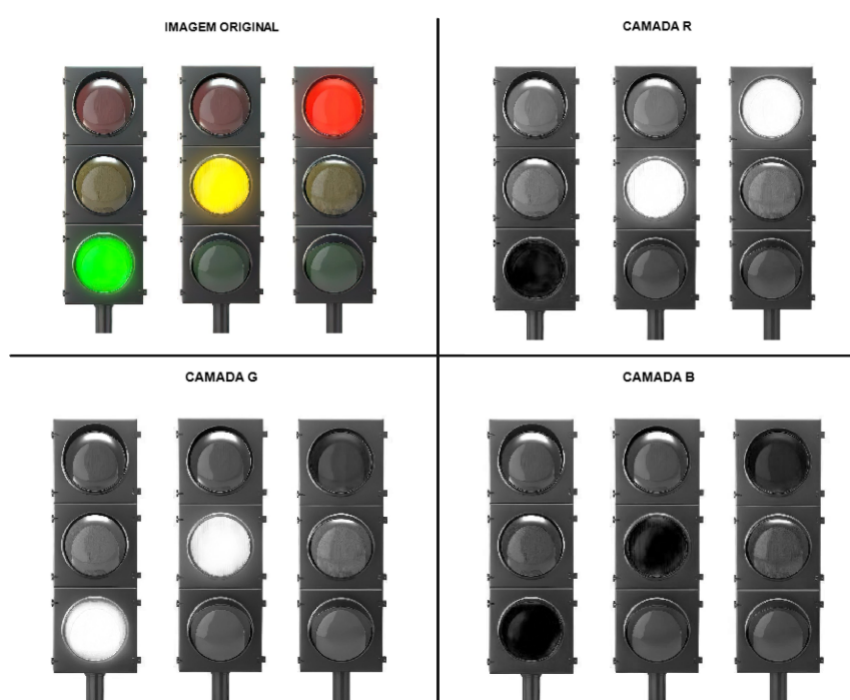
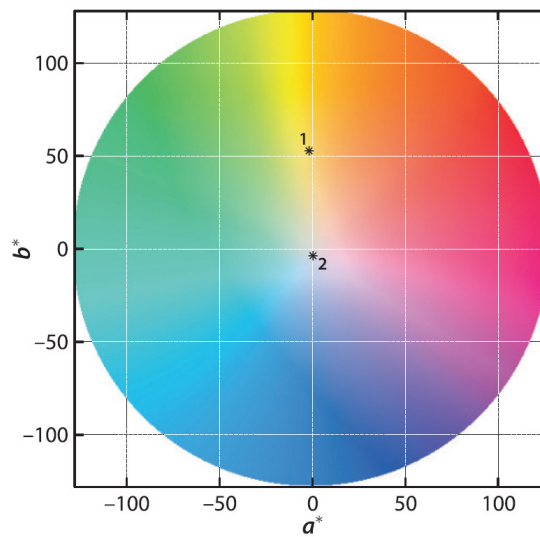


Figura 4 – Decomposição da imagem original nas camadas R, G e B.

Fonte – Fonte autoral

Outro espaço de cor bastante utilizado para representação de imagens coloridas é o CIELAB, o qual foi definido em 1976 pela International Commission on Illumination (CIE), de modo que uma variação numérica nos valores dos pixels correspondesse à uma variação proporcional na percepção da cor. No espaço de cor CIELAB a primeira camada caracteriza a variação do brilho da imagem, a segunda representa a variação de verde para vermelho, por fim, a terceira camada denota a variação do azul para o amarelo [10]. Para fins de ilustração, o plano *ab* do espaço de cor CIELAB é apresentado na Figura 5.

Na Figura 6 são apresentadas as três camadas de uma imagem representada no espaço de cor CIELAB. Nesta figura, observa-se que na camada A, cujos valores denotam a variação da cor verde para o vermelho, a região em verde é escura, enquanto a região em

Figura 5 – Plano  $ab$  do espaço de cor CIELAB.

Fonte – Fonte: [6]

vermelho é a mais clara da imagem (por apresentar os valores mais elevados). Na camada L, a qual representa a intensidade do brilho, as cores verde e amarelo possuem maior destaque devido ao alto brilho nesta cena, sendo o amarelo mais presente que o verde. Por outro lado, na camada B (representa a variação do azul para o amarelo) o amarelo possui maior destaque, seguido pelo verde e com menor notoriedade pela cor vermelha.

## 2.3 Processamento digital de imagens

Processamento digital de imagem consiste na aplicação de técnicas/algoritmos para capturar, manipular e transformar imagens com o auxílio de ferramentas computacionais. A aplicação dessas técnicas permite realizar a análise dos elementos presentes na imagem, auxiliando na percepção humana e no processo de automação em aplicações industriais [13].

Especificamente, a análise de uma imagem tem como objetivo adquirir uma descrição que contemple as informações pertinentes para realizar a distinção dos elementos de interesse presentes em uma cena [14]. Estas informações podem ser baseadas na textura, cor, forma, contraste ou ainda nos níveis de cinza de um objeto, por exemplo.

Nas subseções seguintes são discutidas algumas técnicas de processamento digital de imagem utilizadas no desenvolvimento deste trabalho. Inicialmente são abordadas a segmentação de objetos baseados em cor, operações morfológicas, análise de componentes conectados, seguido pela extração de características de regiões e, por fim, são abordados conceitos relativos ao rastreamento de objetos utilizando filtro de Kalman.

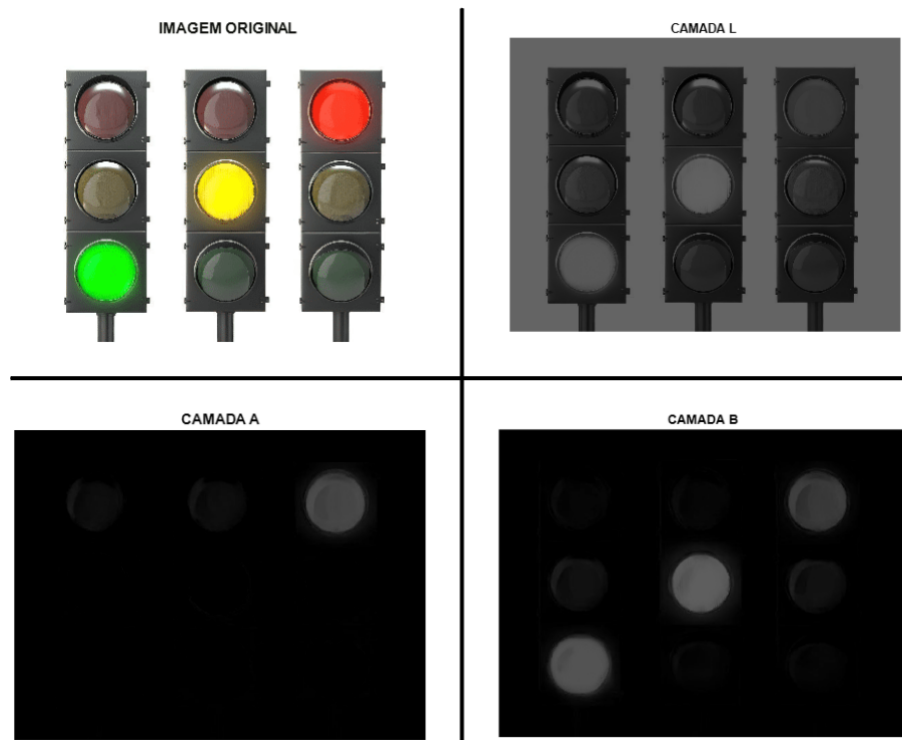


Figura 6 – Decomposição da imagem original nas camadas L, A e B.

Fonte – Fonte autoral

### 2.3.1 Segmentação de objetos baseado em cor

Uma maneira de identificar e diferenciar um objeto de interesse dos demais presentes em uma cena é por meio da sua cor [15]. Para tal, pode-se utilizar um algoritmo de segmentação de imagem baseado em cor, onde uma imagem de entrada  $\mathbf{I}$  é processada, os pixels com cor próxima a uma cor de referência (definida pelo projetista) são identificados, e uma imagem de saída  $\mathbf{M}$  binária é gerada com suas regiões em branco correspondendo aos objetos de cor similar àquela definida pelo projetista.

A identificação de um objeto de interesse por meio de sua cor demanda aplicação de etapas processuais responsáveis por organizar e filtrar as informações presentes na cena [16]. Na Figura 7 é apresentado um diagrama com as etapas de processamento de um algoritmo de segmentação de imagem baseado em cor. Tomando como ponto inicial uma imagem de entrada  $\mathbf{I}$  no espaço de cor RGB, o primeiro procedimento do algoritmo de segmentação é a transformação dessa imagem para o espaço de cor CIELAB. Em seguida, tendo-se em mãos a cor de referência do objeto de interesse no espaço de cor CIELAB (com componentes  $L_r$ ,  $a_r$  e  $b_r$ ) obtém-se a imagem de distância  $\mathbf{D}$ , com cada um dos seus pixels sendo computados como:

$$\mathbf{D}(u, v) = \sqrt{[\mathbf{L}(u, v) - L_r]^2 + [\mathbf{a}(u, v) - a_r]^2 + [\mathbf{b}(u, v) - b_r]^2} \quad (2.1)$$

onde  $\mathbf{L}(u, v)$ ,  $\mathbf{a}(u, v)$  e  $\mathbf{b}(u, v)$  correspondem aos pixels de coordenada  $(u, v)$  das camadas

$\mathbf{L}$ ,  $\mathbf{a}$  e  $\mathbf{b}$ , respectivamente, da imagem de entrada no espaço de cor CIELAB. Observe que cada pixel com coordenada  $(u, v)$  da imagem de distância  $\mathbf{D}$  é determinado pela distância Euclidiana entre o pixel da imagem de entrada no espaço de cor CIELAB e a cor de referência. Assim, quanto menor for o valor de  $\mathbf{D}(u, v)$ , maior será a proximidade entre a cor do pixel analisado e a cor de referência.

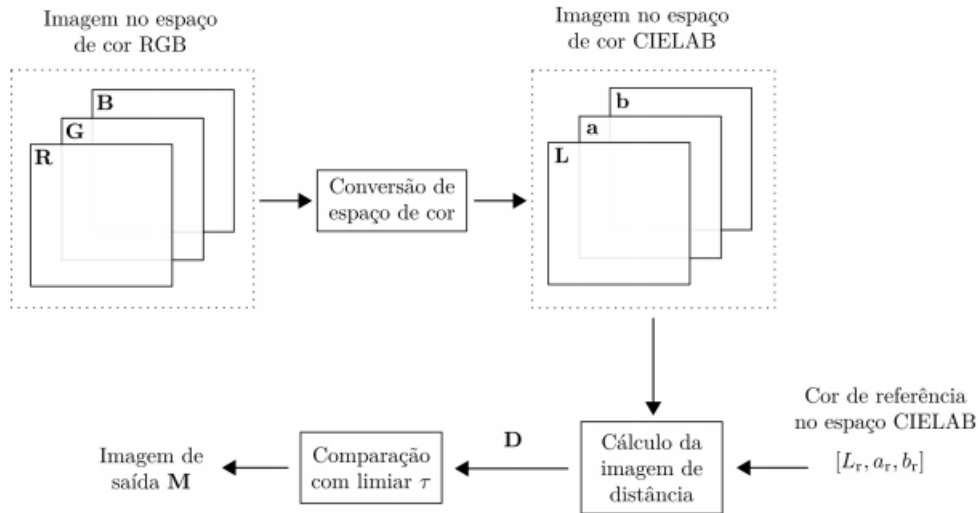


Figura 7 – Etapas de processamento do algoritmo de segmentação de imagem baseado em cor.

Fonte – [17]

Finalmente, a imagem binária  $\mathbf{M}$  é obtida a partir da comparação de cada pixel da imagem  $\mathbf{D}$  com um limiar  $\tau$  (também definido pelo projetista), da seguinte forma:

$$\mathbf{M} = (\mathbf{D} < \tau). \quad (2.2)$$

Desta forma, as regiões em branco da imagem  $\mathbf{M}$  correspondem às regiões na imagem de entrada com cor próxima à cor de referência.

Um exemplo de aplicação do algoritmo de segmentação de imagem baseado em cor é apresentado na Figura 8. Note que os valores apresentados na figura estão no espaço de cor RGB, entretanto esses valores são apresentados apenas como base para a identificação da cor vermelha. Os valores reais de referência para o restante do processo foram obtidos no espaço de cor CIELAB.

### 2.3.2 Operações morfológicas

As operações morfológicas são técnicas de processamento espacial não lineares utilizadas para modificar a forma de regiões presentes em uma imagem binária  $\mathbf{I}$ , a partir de um elemento estruturante  $\mathbf{S}$ . Especificamente, um elemento estruturante  $\mathbf{S}$  consiste também em uma imagem binária (usualmente, com dimensões menores do que a imagem  $\mathbf{I}$  a ser



Figura 8 – Exemplo de resultado obtido a partir da segmentação da imagem, tendo o vermelho como cor de referência.

Fonte – Fonte autoral

processada), contendo um pixel de referência. Na Figura 9 são apresentados exemplos de elementos estruturantes, com indicação do pixel de referência (em cinza). Nessa imagem, apesar do pixel de referência ser apresentado em cinza, é importante destacar que o seu valor é igual 1.

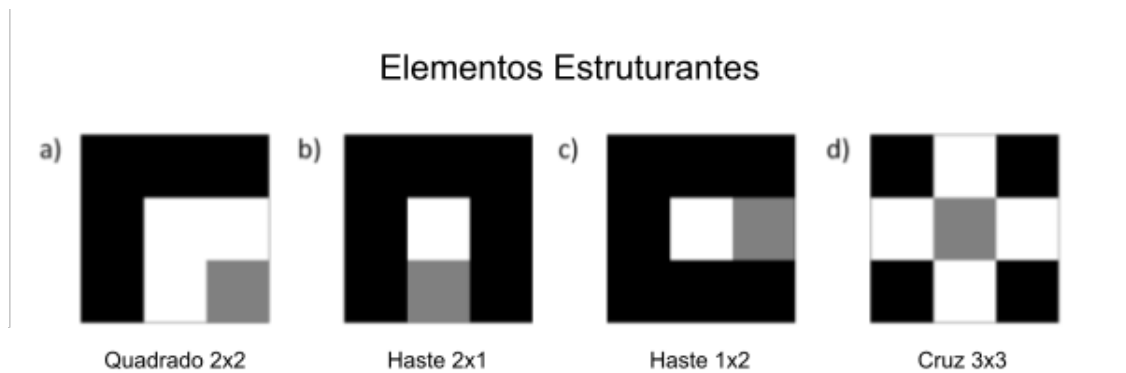


Figura 9 – Exemplos de elementos estruturantes.

Fonte – [18]

Em particular, operações morfológicas são bastante utilizadas para remoção de imperfeições em imagens binárias, visando por exemplo uma melhor definição das bordas de regiões de interesse [19]. Um exemplo de resultado que pode ser obtido por meio de uma operação morfológica é apresentado na Figura 10.

Em uma operação morfológica, o valor de cada pixel  $(u, v)$  na imagem de saída  $\mathbf{O}$  é determinado com base na análise de uma subregião na imagem de entrada  $\mathbf{I}$  determinada por um elemento estruturante  $\mathbf{S}$ . Matematicamente, tem-se

$$\mathbf{O}(u, v) = f[\mathbf{I}(u + i, v + j)], \quad \forall (i, j) \in \mathbf{S} \text{ e } (u, v) \in \mathbf{I} \quad (2.3)$$

onde  $f(\cdot)$  é uma função não linear que define o tipo de operação morfológica. Especificamente, em uma operação morfológica o elemento estruturante  $\mathbf{S}$  percorre a imagem de entrada  $\mathbf{I}$ , gerando em cada etapa do processamento um pixel na imagem resultante

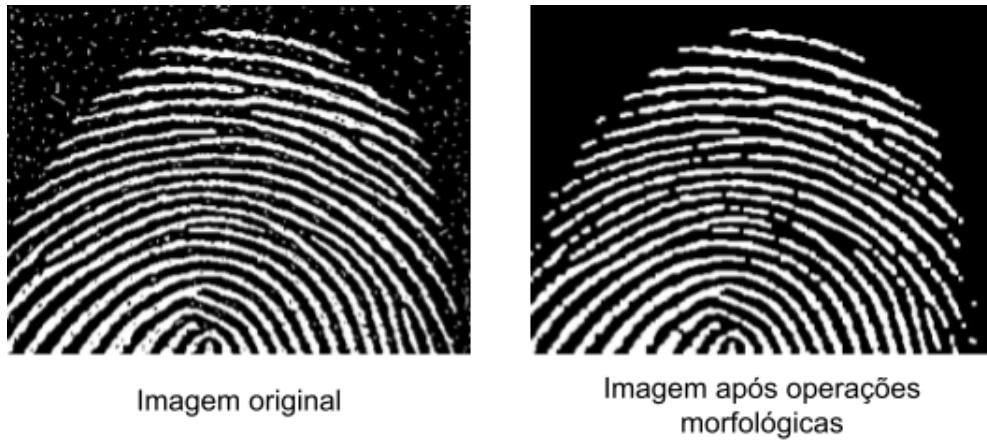


Figura 10 – Exemplo de resultado obtido por meio de operações morfológicas.

Fonte – [20]

$O$  na coordenada  $(u, v)$  correspondente a localização do pixel de referência do elemento estruturante. O valor do pixel gerado em cada etapa do processamento depende: 1) da existência ou não de sobreposição entre o elemento estruturante  $S$  e a região de interesse da imagem de entrada  $I$ ; e 2) do tipo de operação morfológica conduzida (definida pela função  $f(\cdot)$ ).

Existem dois tipos fundamentais de operações morfológicas: erosão e dilatação [21]. A primeira consiste na aplicação de um elemento estruturante  $S$  que, apenas ao sobrepor totalmente uma região de interesse na imagem de entrada  $I$ , o pixel de referência resultante na imagem de saída  $O$  recebe o valor um. Ou seja, quanto maior for  $S$ , menor será o resultado da erosão na imagem de saída. Matematicamente, a operação de erosão é representada por:

$$O = I \ominus S \quad (2.4)$$

Um exemplo da aplicação da operação morfológica de erosão é apresentada na Figura 11, onde nota-se que na imagem de saída  $O$  os pixels com valor 1 (representados na imagem em cinza) são aqueles em que ocorre sobreposição total entre o elemento estruturante  $S$  e a correspondente região na imagem de entrada  $I$ .

Diferentemente da erosão, a dilatação consiste na aplicação de um elemento estruturante  $S$  que, ao sobrepor parcialmente qualquer região de interesse na imagem de entrada  $I$ , o pixel de referência resultante na imagem de saída  $O$  recebe o valor um. Ou seja, quanto maior o tamanho de  $S$ , maior será a região dilatada na imagem de saída. A equação que representa a operação de dilatação é dada por:

$$O = I \oplus S \quad (2.5)$$

A partir das operações fundamentais de erosão e dilatação podem ser obtidas as opera-



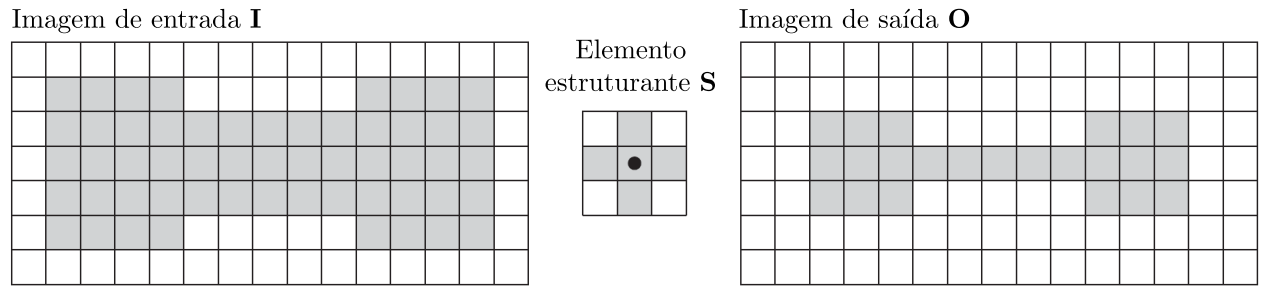


Figura 11 – Exemplo de operação morfológica de erosão.

Fonte – Adaptado de [12]

ções morfológicas de abertura e fechamento [22]. Assim, a operação de abertura consiste inicialmente na implementação da erosão e, na sequência, a dilatação. Tal operação, tem como característica a eliminação de pequenas regiões em branco na imagem de entrada, preservando a forma/dimensões das regiões maiores. Matematicamente, a operação de abertura é representada como:

$$O = (I \ominus S) \oplus S = I \circ S \quad (2.6)$$

A operação de fechamento é representada pela aplicação inversa dos processos implementados na abertura, ou seja, primeiramente é realizada a dilatação e, em seguida, a erosão. Especificamente, tal operação tem por objetivo preencher pequenas lacunas/buracos em regiões de interesse, preservando o formato da parte externa da região. A equação que representa a operação de fechamento é dada por:

$$O = (I \oplus S) \ominus S = I \bullet S \quad (2.7)$$

Na Figura 12 é apresentada uma imagem de entrada **I** e o resultado de todas as operações morfológicas descritas anteriormente.

### 2.3.3 Análise de componentes conectados

Em uma imagem binária as regiões em branco são comumente chamadas de componentes conectados. Para diferenciar os componentes conectados de uma imagem binária, pode-se utilizar um algoritmo de análise de componentes conectados, onde atribui-se valores inteiros (rótulos) aos pixels de cada região de interesse [24]. Desta forma, após tal procedimento, obtém-se uma imagem rotulada **R** com seus pixels possuindo valores que variam de zero (regiões que são irrelevantes nesse contexto) até o número de componentes conectados identificados, como apresentado na Figura 13.

A Figura 14 representa a diferenciação dos componentes conectados por meio da coloração na escala de cinza, onde o primeiro objeto de interesse está localizado mais próximo do vértice esquerdo superior. O segundo objeto está representado pelo nível intermediário

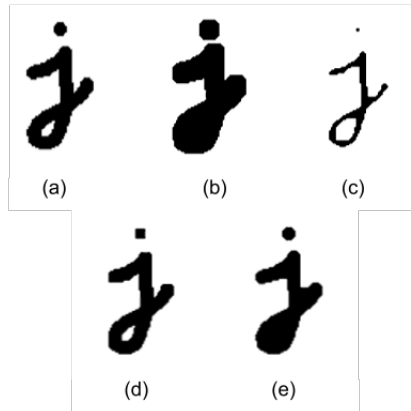


Figura 12 – Operações morfológicas: (a) Imagem de Entrada; (b) Dilatação; (c) Erosão; (d) Abertura; (e) Fechamento.

Fonte – [23]

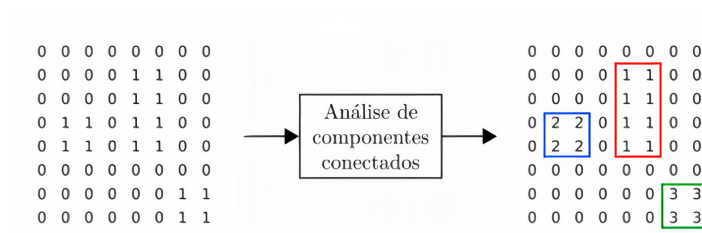


Figura 13 – Resultado da análise de componentes conectados.

Fonte – [25]

na escala de cinza, localizado mais próximo ao vértice esquerdo inferior. Por fim, o último objeto de interesse descrito possui cor branca e está mais próximo da aresta limitante à direita da imagem.

Especificamente, a partir da imagem rotulada  $\mathbf{R}$  é possível separar os componentes conectados identificados em imagens binárias separadas (veja Figura 15). Por exemplo, para obter uma imagem binária  $\mathbf{I}_1$  contendo apenas o componente conectado com rótulo 1, faz-se a seguinte operação de comparação lógica:

$$\mathbf{I}_1 = (\mathbf{R} == 1). \tag{2.8}$$

### 2.3.4 Extração de características de região

No contexto de processamento digital de imagem, características de região referem-se a valores numéricos (usualmente, representados no formato de escalares ou vetores) que

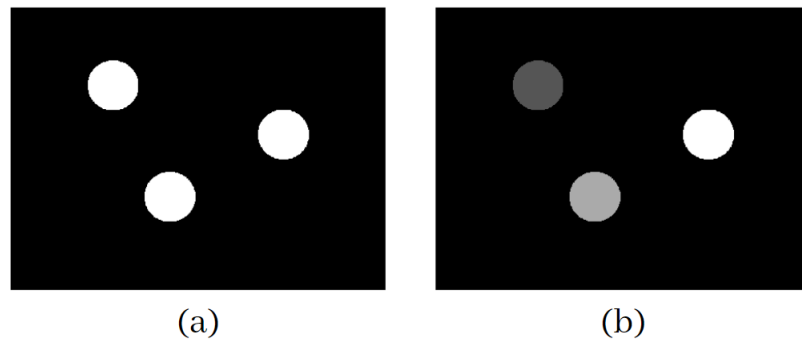


Figura 14 – Análise de componentes conectados. (a) Imagem de entrada; (b) Componentes conectados.

Fonte – Fonte autoral

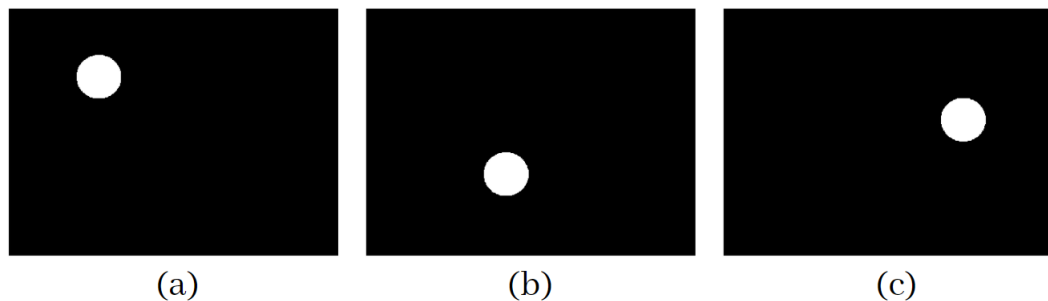


Figura 15 – Componentes conectados apresentados separadamente. (a) Objeto 1; (b) Objeto 2; (c) Objeto 3.

Fonte – Fonte autoral

podem ser utilizados para caracterizar a posição, orientação e o formato de uma dada região de interesse em uma imagem binária. Dentre as características de regiões, pode-se citar:

1. *Bounding box*: menor retângulo com bordas paralelas aos eixos da imagem que contém a região de interesse [26]. Sua dimensão pode ser determinada a partir das coordenadas dos vértices superior esquerdo e inferior direito do próprio *bounding box*. Desta forma, o cálculo para a obtenção destas coordenadas se baseia na aquisição das coordenadas de linha e coluna de todos os pixels presentes na região de interesse, sendo a coordenada superior esquerda do *bounding box* composta pela coordenada de coluna do pixel mais a esquerda e pela coordenada de linha do pixel mais acima. De forma semelhante, a coordenada inferior direita do *bounding box* é composta pela coordenada de coluna do pixel mais a direita e pela coordenada de linha do pixel mais abaixo.

A Figura 16 apresenta um objeto de interesse com a forma geométrica de um triângulo com o seu *bounding box* com linhas na cor verde.

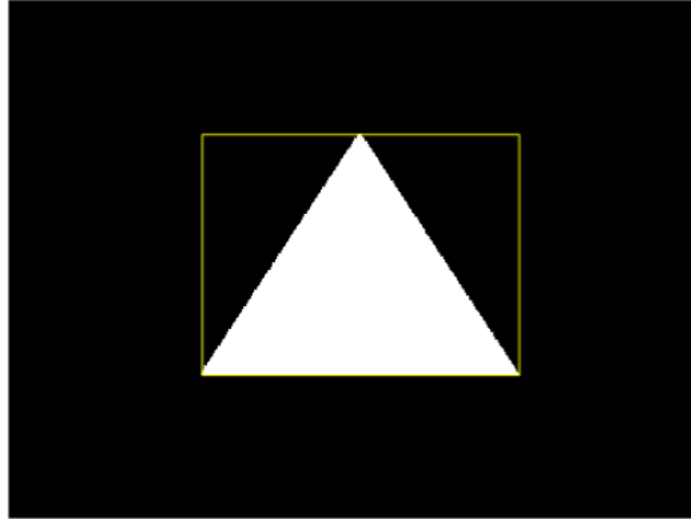


Figura 16 – Exemplo de *bounding box* em torno de uma região de interesse.

Fonte – Fonte autoral

2. *Momentos*: são métricas utilizadas para aferir a distribuição dos pixels de uma região de interesse (em uma imagem binária) nas direções horizontal e vertical [26], sendo calculados como:

$$m_{pq} = \sum u^p \cdot v^q I(u, v) \quad (2.9)$$

Os valores de  $p$  e  $q$  são parâmetros da função que definem a ordem do momento, especificamente, define-se como a ordem do momento o resultado de  $p + q$ . Adicionalmente,  $u$  e  $v$  denotam as coordenadas de cada pixel da região de interesse, sendo a primeira correspondente à posição da coluna e a segunda correspondente à posição de linha do pixel. Assim, o momento de ordem zero será obtido quando  $p$  e  $q$  assumirem valor zero, e o resultado da equação será correspondente ao valor da área do objeto de interesse.

O cálculo dos momentos de ordem um é realizado em duas etapas, sendo que na primeira etapa é atribuído o valor 1 para o parâmetro  $p$  e o valor 0 para  $q$ . A segunda etapa consiste na inversão dos valores, ou seja,  $p = 0$  e  $q = 1$ . Desta forma, são obtidos dois valores para o momento de ordem um, onde o primeiro valor obtido está associado à distribuição dos pixels da região de interesse em relação ao eixo horizontal, enquanto o segundo valor está associado à distribuição dos pixels em relação ao eixo vertical.

3. *Centroide*: corresponde as coordenadas do centro geométrico de uma região [26], sendo obtido a partir da divisão dos momentos de ordem um pelos momentos de

ordem zero de uma região de interesse, como apresentado a seguir:

$$u_c = \frac{m10}{m00} \quad (2.10)$$

e

$$v_c = \frac{m01}{m00} \quad (2.11)$$

onde  $u_c$  é a coordenada de coluna do centroide e  $v_c$ , a coordenada de linha. Assim, a partir da obtenção das coordenadas do centroide, tem-se informações suficientes para realizar o mapeamento de uma região em relação à origem da imagem. Na Figura 17 o centroide do triângulo é identificado com um asterisco azul, com suas coordenadas sendo obtidas a partir dos cálculos descritos anteriormente. Nota-se que o centro geométrico do objeto de interesse é simétrico à todos os pontos do triângulo pelo fato dele ser equilátero.

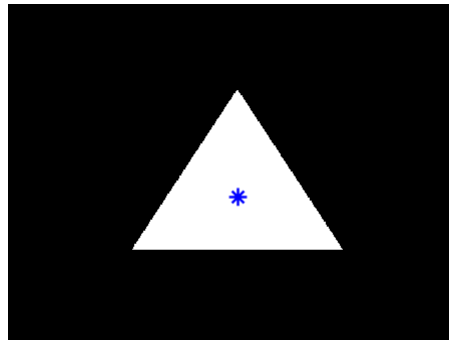


Figura 17 – Centroide do objeto de interesse.

Fonte – Fonte autoral

## 2.4 Filtro de Kalman

O filtro de Kalman é um algoritmo matemático proposto por Rudolf Kalman em 1960, para estimação em tempo real do estado de um sistema dinâmico. O filtro de Kalman pode ser utilizado quando algumas (ou todas) as variáveis de estado do sistema não podem ser medidas diretamente e/ou quando as medições estão contaminadas por ruído [27]. Para tal, o filtro de Kalman opera em duas etapas, a saber: predição e estimação. Na etapa de predição o estado do sistema de interesse é predito com base no modelo dinâmico utilizado para representação do sistema e, também, com base na última atualização do seu estado. Na etapa seguinte, o estado do sistema é estimado (refinando o resultado) a partir da predição da etapa anterior e das medidas obtidas diretamente do sistema [4].

O filtro Kalman tem sido utilizado nas últimas décadas em diversas aplicações práticas, dentre as quais pode-se citar o rastreamento de objetos [28]. Neste caso, o objetivo é: a partir de medições ruidosas e/ou descontínuas da posição do objeto de interesse obter

uma estimativa mais precisa da sua posição. Com objetivo de ilustrar a aplicação do filtro de Kalman no rastreamento de objetos, considere um objeto se movimentando em um plano com coordenadas  $(u, v)$ , conforme ilustrado na Figura 18. Assuma que em cada instante de tempo são obtidas medidas ruidosas da posição desse objeto, sendo tais medidas armazenadas em um vetor  $\mathbf{z}_k$ , onde  $k$  denota o instante de tempo:

$$\mathbf{z}_k = \begin{bmatrix} u_k \\ v_k \end{bmatrix} \quad (2.12)$$

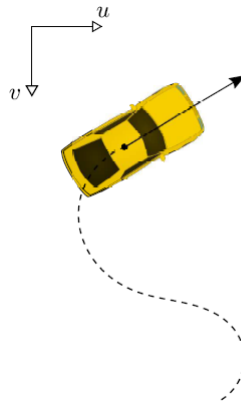


Figura 18 – Objeto se movimentando no plano  $(u, v)$ .

Fonte – [29]

Neste caso, pode-se utilizar o filtro de Kalman para obter valores mais precisos da posição do objeto. Para tal, primeiramente, define-se um modelo em espaço de estados representando a dinâmica do objeto. Assuma um objeto que possui velocidade constante e aceleração nula (em um intervalo  $\Delta T$ ). As equações que descrevem sua posição e velocidade são, respectivamente:

$$x_k = x_{k-1} + \Delta T \cdot \dot{x}_{k-1} \quad (2.13)$$

$$\dot{x}_k = \dot{x}_{k-1} \quad (2.14)$$

onde  $k$  e  $k - 1$  representam, respectivamente, os instantes de tempo atual e anterior. De forma mais detalhada, o vetor estado  $\mathbf{x}_k$ , que descreve a posição e velocidade do objeto em um espaço bidimensional, é dado por:

$$\mathbf{x}_k = \begin{bmatrix} u_k \\ \dot{u}_k \\ v_k \\ \dot{v}_k \end{bmatrix} \quad (2.15)$$

onde  $u_k$  e  $v_k$  denotam a posição do objeto em relação aos eixos horizontal e vertical, com  $\dot{u}_k$  e  $\dot{v}_k$  representando a velocidade. Assim, ao representar o sistema completo em um modelo em espaço de estados, tem-se:

$$\begin{bmatrix} u_k \\ \dot{u}_k \\ v_k \\ \dot{v}_k \end{bmatrix} = \begin{bmatrix} 1 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_{k-1} \\ \dot{u}_{k-1} \\ v_{k-1} \\ \dot{v}_{k-1} \end{bmatrix} \quad (2.16)$$

Representando de forma compacta e considerando que o modelo utilizado não é perfeito, ou seja, assumindo perturbações aleatórias do sistema  $\mathbf{s}_k$  na determinação do estado, tem-se a equação do modelo dinâmico do sistema:

$$\mathbf{x}_k = \mathbf{F} \cdot \mathbf{x}_{k-1} + \mathbf{s}_k \quad (2.17)$$

Para a saída do sistema, tem-se a equação que descreve a posição medida do objeto em cada instante de tempo somada com um ruído aleatório de medição ( $\mathbf{w}_k$ ), assim:

$$\mathbf{z}_k = \mathbf{H} \cdot \mathbf{x}_k + \mathbf{w}_k \quad (2.18)$$

onde

$$\mathbf{F} = \begin{bmatrix} 1 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.19)$$

e

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (2.20)$$

Após a definição do modelo em espaço de estados, pode-se utilizar iterativamente o filtro de Kalman para obter estimativas da posição do objeto. Primeiramente, executa-se a etapa de predição, composta pelas equações:

$$\tilde{\mathbf{x}}_k = \mathbf{F} \cdot \hat{\mathbf{x}}_{k-1} \quad (2.21)$$

e

$$\tilde{\mathbf{P}}_k = \mathbf{F} \cdot \hat{\mathbf{P}}_{k-1} \cdot \mathbf{F}^T + \hat{\mathbf{S}} \quad (2.22)$$

onde  $\tilde{\mathbf{x}}_k$  é o vetor de estado predito no instante atual e  $\hat{\mathbf{x}}_{k-1}$  representa o vetor de estado estimado no instante de tempo anterior. Além disso,  $\tilde{\mathbf{P}}_k$  é a matriz de covariância do estado predito no instante  $k$  e  $\hat{\mathbf{P}}_{k-1}$  é a matriz de covariância do estado estimado no

instante anterior. Essas matrizes de covariância modelam as incertezas nas predições e estimativas do vetor de estado. Em particular, para a matriz  $\hat{\mathbf{S}}$  utiliza-se

$$\hat{\mathbf{S}} = \begin{bmatrix} \Delta T^4/4 & \Delta T^3/2 & 0 & 0 \\ \Delta T^3/2 & \Delta T^2 & 0 & 0 \\ 0 & 0 & \Delta T^4/4 & \Delta T^3/2 \\ 0 & 0 & \Delta T^3/2 & \Delta T^2 \end{bmatrix} \sigma_s^2 \quad (2.23)$$

onde  $\sigma_s^2$  é um parâmetro definido pelo projetista que modela a incerteza do modelo dinâmico escolhido para representação do sistema. Quanto maior valor de  $\sigma_s^2$  assume-se que menos preciso é o modelo, implicando que o vetor de estado predito  $\tilde{\mathbf{s}}_k$  terá menor importância na formação do vetor de estado estimado  $\hat{\mathbf{x}}_k$  determinado na próxima etapa.

Adicionalmente, como condições iniciais para os cálculos da etapa de predição utiliza-se

$$\hat{\mathbf{P}}_0 = \mathbf{I} \quad (2.24)$$

e

$$\hat{\mathbf{x}}_0 = \mathbf{H}^T \cdot \mathbf{z}_0 \quad (2.25)$$

onde  $\mathbf{z}_0$  denota a posição inicial medida para o objeto.

Na sequência (isto é, após a etapa de predição), conduz-se a etapa de estimação. A execução dessa etapa é composta por quatro equações, iniciando pelo cálculo da diferença entre a posição do objeto medida (por algum sensor) e a estimativa da posição calculada na etapa anterior (veja Equação (2.26)). Na sequência, realiza-se o cálculo do ganho de Kalman  $\mathbf{K}_k$ , do vetor de estado estimado  $\hat{\mathbf{x}}_k$  e, finalmente, da matriz de covariância estimada  $\hat{\mathbf{P}}_k$ , conforme (2.27), (2.28) e (2.29), respectivamente,

$$\mathbf{e}_k = \mathbf{z}_k - \mathbf{H} \cdot \tilde{\mathbf{x}}_k \quad (2.26)$$

$$\mathbf{K}_k = \tilde{\mathbf{P}}_k \cdot \mathbf{H}^T \cdot (\mathbf{H} \cdot \tilde{\mathbf{P}}_k \cdot \mathbf{H}^T + \hat{\mathbf{W}})^{-1} \quad (2.27)$$

$$\hat{\mathbf{x}}_k = \tilde{\mathbf{x}}_k + \mathbf{K}_k \cdot \mathbf{e}_k \quad (2.28)$$

$$\hat{\mathbf{P}}_k = \tilde{\mathbf{P}}_k - \mathbf{K}_k \cdot \mathbf{H} \cdot \tilde{\mathbf{P}}_k \quad (2.29)$$

onde

$$\hat{\mathbf{W}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \sigma_w^2 \quad (2.30)$$

sendo  $\sigma_w^2$  um parâmetro definido pelo projetista, para aferir o nível de incerteza nas medições obtidas da posição do objeto.



Com isso, a estimativa da posição do objeto em determinado instante de tempo pode ser obtida por meio do vetor estado estimado  $\hat{\mathbf{x}}_k$  como

$$\hat{\mathbf{z}}_k = \mathbf{H} \cdot \hat{\mathbf{x}}_k \quad (2.31)$$

onde  $\hat{\mathbf{z}}_k$  denota o vetor contendo a posição estimada do objeto no instante  $k$ . Vale destacar que em alguns instantes de tempo as posições medidas  $\mathbf{z}_k$  do objeto podem não ser obtidas por algum motivo (falha na medição) e, nesse caso, os cálculos são realizados parcialmente. Isto é, na etapa de predição os cálculos são realizados integralmente, porém na estimação  $\mathbf{e}_k$  não pode ser obtido pois depende do vetor de posições  $\mathbf{z}_k$ . Nesse caso, considera-se que o estado estimado é igual ao estado predito, isto é,

$$\hat{\mathbf{x}}_k = \tilde{\mathbf{x}}_k \quad (2.32)$$

## 3 Desenvolvimento

Este capítulo discute o sistema de rastreamento de múltiplos objetos desenvolvido neste trabalho. Para tal, primeiramente, é apresentada uma descrição geral do sistema e, na sequência, são descritas de forma detalhada cada etapa de processamento.

### 3.1 Sistema proposto

Neste trabalho foi desenvolvido um sistema para rastreamento de múltiplos objetos presentes em cenas de um vídeo. Em particular, no sistema desenvolvido, cada objeto é rastreado por meio de um filtro de Kalman criado exclusivamente para tal objeto.

O sistema de rastreamento é capaz de criar ou remover de forma dinâmica novos filtros de Kalman a medida que objetos surgem ou saem da cena. Para fins de simplificação, é assumido que os objetos presentes na cena possuem cor uniforme conhecida (visando facilitar a etapa de detecção).

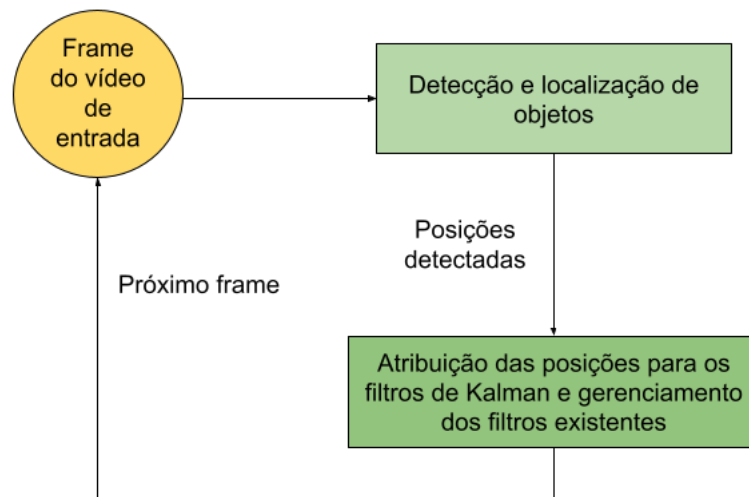


Figura 19 – Fluxograma das etapas de processamento realizadas no sistema de rastreamento desenvolvido.

Fonte – Fonte autoral

O diagrama ilustrando as etapas de processamento do sistema desenvolvido é apresentado na Figura 19. Observa-se que cada quadro do vídeo passa por duas etapas principais de processamento, a saber: detecção e localização do objeto e atribuição das posições para

os filtros de Kalman e gerenciamento dos filtros existentes. Especificamente, o algoritmo de detecção e localização dos objetos de interesse é implementado para realizar a aquisição das informações da posição de cada objeto que são utilizadas como entrada na etapa de predição do(s) filtro(s) de Kalman. Na sequência, é verificada a quantidade de filtros de Kalman existentes e a quantidade de objetos de interesse detectados, sendo realizada uma atribuição das posições medidas aos filtros de Kalman existentes. Caso existam posições não atribuídas a filtros existentes, realiza-se o gerenciamento, onde é realizada a criação de novos filtros ou, ainda, a exclusão de filtros existentes que passarão a ser inativos. Todo o processo descrito se repete até que o último frame do vídeo seja identificado e processado.

### 3.1.1 Detecção e localização de objetos

Nesta seção é descrito o algoritmo de detecção e localização de objetos utilizado no sistema de rastreamento desenvolvido. Visando simplificar esta etapa, assume-se que os objetos presentes nos vídeos analisados possuem cor uniforme e conhecida. Em particular, vale destacar que o procedimento apresentado nesta seção é aplicado para cada quadro do vídeo em análise.

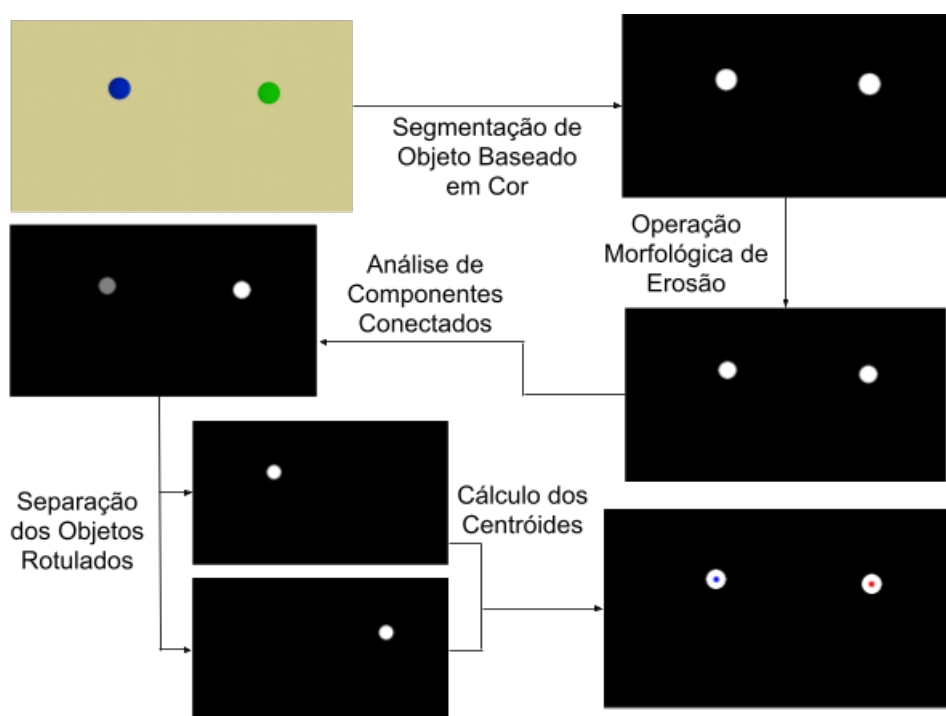


Figura 20 – Diagrama do processo de detecção e localização de objetos.

Fonte – Fonte autoral

Como apresentado na Figura 20, o frame de entrada é inicialmente processado a partir do algoritmo de segmentação de imagem baseado em cor, com o propósito inicial de obter uma imagem binária com as regiões em branco correspondendo aos objetos de interesse.

Na sequência, a operação morfológica de erosão é implementada para remover pequenas regiões em branco (ruídos) que podem surgir do processo de segmentação anterior. Na próxima etapa do processo, realiza-se a análise de componentes conectados visando separar os objetos de interesse em imagens binárias distintas. Na sequência, para cada imagem binária, obtém-se o centroide de cada objeto. Desta forma, as informações obtidas são suficientes para a próxima etapa de processamento, sendo utilizadas como entrada para os filtros de Kalman.

### 3.1.2 Algoritmo de atribuição das posições e gerenciamento dos filtros de Kalman

Esta etapa do processamento consiste em atribuir aos filtros de Kalman existentes às posições medidas dos objetos de interesse em cada frame do vídeo e, também, em executar as etapas de predição e estimação de cada filtro de Kalman criado. Na sequência, é realizado o gerenciamento dos filtros de acordo com a demanda em cada frame.

Para fins de exemplificação, considere inicialmente que apenas um objeto de interesse será identificado durante o vídeo e que nenhum filtro de Kalman está inicializado. Neste caso, em cada frame do vídeo é verificada a existência de objetos de interesse (como descrito na Seção 3.1.1). Assim, quando um objeto de interesse é identificado na cena, um filtro de Kalman é inicializado a partir das equações (2.24) e (2.25), e seu *status* é definido como ativo.

Em seguida, no próximo frame do vídeo em que o objeto é identificado (tendo sua posição medida), passa-se a executar as etapas de predição e estimação do filtro de Kalman. Primeiramente, realiza-se a predição do vetor de estados através das equações (2.21) e (2.22). Em seguida, utilizando a informação da nova posição medida do objeto, as equações para a estimação do vetor de estado são aplicadas (isto é, (2.26), (2.27), (2.28) e (2.29)). Com as coordenadas da posição estimada do objeto de interesse sendo obtidas a partir da equação (2.31). Em particular, esse processamento é repetido para cada um dos frames do vídeo.

No processamento executado, vale destacar que caso o filtro referente ao objeto de interesse esteja com *status* ativo e em certo frame do vídeo a posição deste objeto não seja medida (devido a uma oclusão momentânea, por exemplo), somente a etapa de predição é calculada, sendo aplicada a equação (2.32) para a estimação.

Considere agora o caso mais geral de um vídeo com múltiplos objetos na cena. Neste caso, a relação entre o número de filtros de Kalman existentes e a quantidade de posições medidas pode variar de acordo com o contexto do frame analisado. Por exemplo, dependendo do frame pode existir mais filtros de Kalman ativos do que posições medidas (devido, por exemplo, a oclusão momentânea de algum objeto). Ou ainda, podem haver mais posições medidas do que filtros de Kalman (o que pode ocorrer quando um novo ob-

jeto surge na cena, por exemplo). Assim, no caso mais geral, executa-se os processamentos apresentados no fluxograma da Figura 21.

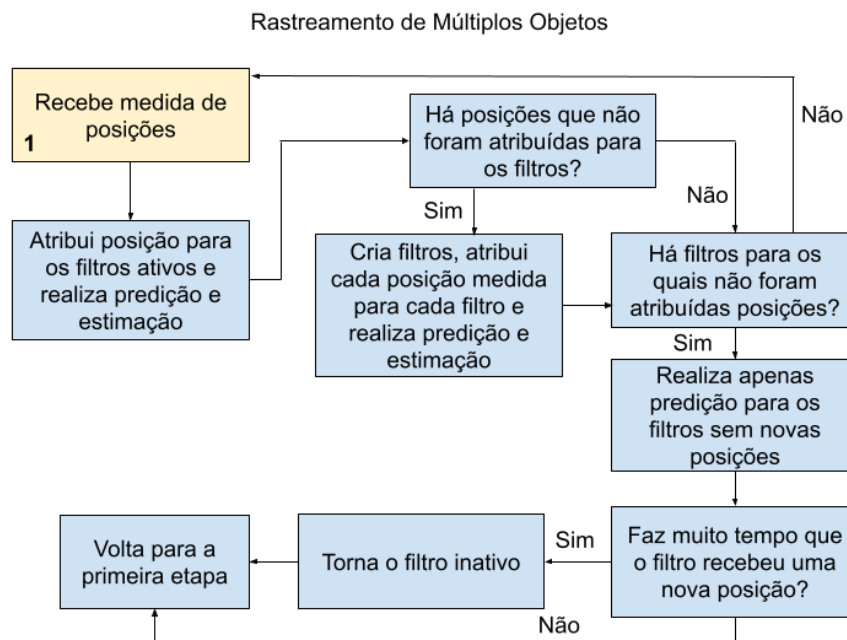


Figura 21 – Fluxograma com atribuição das posições medidas e gerenciamento dos filtros de Kalman.

Fonte – Fonte autoral

Assim, conforme indicado na Figura 21, as posições medidas dos objetos de interesse em cada frame são as informações de entrada do algoritmo. Especificamente, na primeira etapa é realizado o processo de atribuição das posições medidas aos filtros de Kalman. É importante ressaltar que na etapa de atribuição de posições aos filtros de Kalman é realizado utilizando o método Húngaro [28]. Este método tem como objetivo otimizar as combinações existentes para a obtenção do menor custo possível. Sua aplicabilidade no contexto descrito tem a função de identificar a menor distância entre as posições medidas e as posições estimados pelos filtros de Kalman existentes.

Ao receber sua nova posição medida, executa-se para o filtro de Kalman as etapas de predição e estimação, conforme descrito anteriormente. Após isso é necessário avaliar a quantidade de filtros existentes e o número de posições medidas, de forma que para a situação em que haja mais posições medidas do que filtros existentes, novos filtros são criados até que essas duas variáveis possuam o mesmo valor.

Com todas as posições medidas atribuídas para todos os filtros existentes, é realizado o processo inverso, sendo avaliada a quantidade de filtros existentes e excedentes, ou seja, filtros ativos para os quais não foram atribuídos posições. Este processo é importante para situações em que há diminuição no número de posições medidas de um frame para outro.

Etapa do Algoritmo	Condição	Ação
Há posições que não foram atribuídas para os filtros de Kalman.	O número de posições medidas é menor do que o número de filtros de Kalman existentes.	Criação de novos filtros até que a quantidade destes seja igual ao número de posições medidas.
Há filtros de Kalman para os quais não foram atribuídas posições.	O número de posições medidas é maior do que o número de filtros de Kalman existentes.	Realização de predição para filtros excedentes e análise da última vez que estes filtros receberam novas atribuições de posições medidas.

Tabela 1 – Procedimentos adotados para situações em que a quantidade de posições medidas é diferente do número de filtros de Kalman existentes.

Fonte – Fonte autoral

Quando são indentificados filtros em excesso, estes são submetidos à dois processos em sequência antes de torná-los inativos. O primeiro consiste na realização apenas da etapa de predição destes filtros (tendo em vista que as posições medidas não são fornecidas e a situação se enquadra em (2.32)). O segundo processo avalia o tempo desde a última atualização destes filtros com posições medidas associadas a cada um deles. Esta etapa é fundamental para a exclusão de filtros excedentes, sendo que a variável de controle deste processo é o período sem novas atribuições de medidas aos filtros em excesso. Assim, caso o tempo decorrido desde a última atribuição seja superior ao tempo limite definido pelo projetista, os filtros remanescentes são desativados, atualizando os seus *status* para inativos.

A Tabela 1 resume os procedimento adotados quando o número de filtros existentes é diferente do número de posições medidas, além do procedimento realizado para cada uma das condições.

## 4 Resultados

Neste capítulo são apresentados alguns resultados obtidos com o sistema de rastreamento de múltiplos objetos implementado no software de simulação numérica Matlab. Para fins de avaliação do desempenho do sistema implementado, foram produzidos 5 vídeos com objetos redondos de cor sólida percorrendo a cena. Na Figura 22 são apresentados quadros do vídeo abordado no Exemplo 4 (Seção 4.4), onde dois objetos de mesma cor entram e saem de cena durante o vídeo. A disposição dos quadros da Figura 22 respeita a ordem dos acontecimentos do vídeo, de modo que no primeiro quadro, mostrado na Figura 22.a, apenas um objeto de interesse está presente na cena. Na sequência, o segundo objeto de interesse entra em cena (Figura 22.b) e os dois percorrem suas trajetórias (Figura 22.c). Por fim, no último quadro (Figura 22.d) o primeiro objeto não está mais visível e o segundo começa a sair de cena. Todos os vídeos utilizados para validação do sistema de rastreamento foram produzidos utilizando o software de modelagem e animação Blender [30].

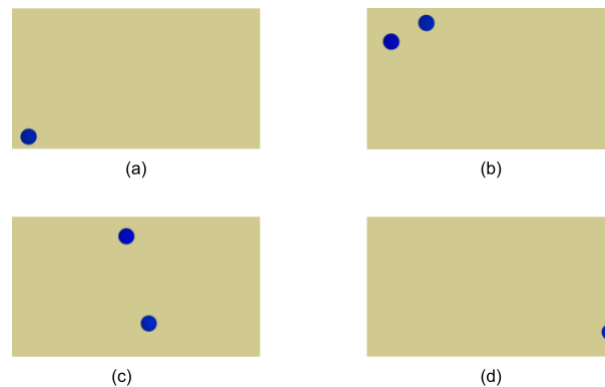


Figura 22 – Quadros do vídeo utilizado no Exemplo 4. (a) tempo = 3.16 s; (b) tempo = 6.75 s; (c) tempo = 10.37 s; (d) tempo = 16.12 s.

Fonte – Fonte autoral

Especificamente, para os resultados apresentados neste capítulo, foram utilizados os valores  $\sigma_w^2 = 1$  e  $\sigma_s^2 = 1$  na parametrização dos filtros de Kalman. A aplicabilidade deste parâmetros é apresentada nas equações (2.23) e (2.30).

### 4.1 Exemplo 1

Este primeiro exemplo tem por objetivo avaliar o desempenho do sistema de rastreamento para o cenário em que apenas um objeto de interesse aparece no vídeo. Neste vídeo em particular, considera-se que o objeto está presente desde o primeiro frame, permanecendo até o último.

Na Figura 23 são apresentados alguns quadros do vídeo utilizado neste exemplo. Observa-se que no primeiro quadro (Figura 23.a) o objeto de interesse está no início de sua trajetória, tendo em vista o tempo decorrido de vídeo. Na sequência, o objeto está realizando sua trajetória e no frame apresentado (Figura 23.b) ele se localiza mais próximo do vértice superior direito da imagem. Por fim é apresentado o último frame do vídeo (23.c), em que o objeto de interesse permanece em cena com seu trajeto concluído.

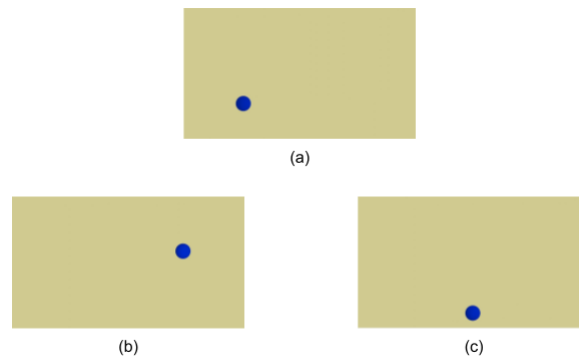


Figura 23 – Quadros do vídeo do Exemplo 1. (a) tempo = 0.25 s; (b) tempo = 8.12 s; (c) tempo = 20.79 s.

Fonte – Fonte autoral

Na Figura 24 são apresentados alguns quadros do vídeo com o resultado do rastreamento do objeto. Nota-se que o sistema foi capaz de rastrear adequadamente o objeto em quadros sucessivos do vídeo. Além disso, conforme indicado, observa-se que apenas 1 filtro de Kalman foi criado para o objeto detectado, indicando o correto funcionamento do sistema. Percebe-se que há uma trajetória em vermelha para cada quadro, a qual representa as últimas 50 posições estimadas pelo filtro de Kalman responsável pelo rastreamento do objeto.

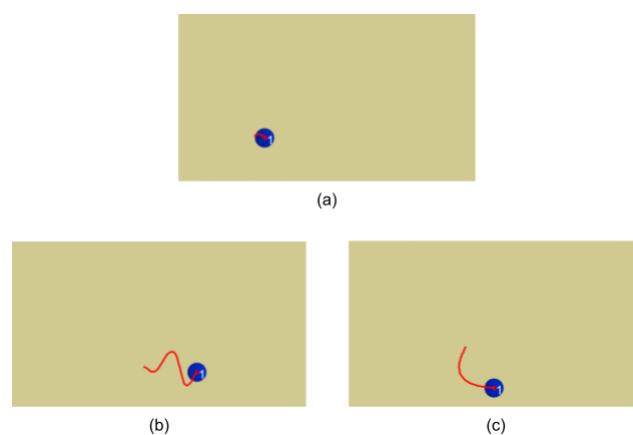


Figura 24 – Quadros do vídeo 1 com rastreamento. (a) tempo = 1.41 s; (b) tempo = 5.99 s; (c) tempo = 20.79 s.

Fonte – Fonte autoral



## 4.2 Exemplo 2

O segundo exemplo tem por objetivo avaliar o desempenho do mesmo sistema de rastreamento para o cenário em que somente um objeto de interesse aparece nas cenas do vídeo. Diferentemente do Exemplo 4.1, o vídeo inicializa sem nenhum objeto presente e durante o vídeo o objeto de interesse entra em cena e percorre seu trajeto, saindo de cena antes do término do vídeo. Assim, nesse cenário é possível avaliar se o sistema de rastreamento é capaz de criar de forma dinâmica um filtro de Kalman assim que um objeto é detectado ao entrar em cena.

A Figura 25 contempla alguns quadros relevantes do vídeo utilizado neste exemplo. Inicialmente, o objeto de interesse está entrando em cena (Figura 25.a). Ao longo do vídeo ele percorre sua trajetória (Figura 25.b) e começa a sair de cena (Figura 25.c) próximo ao término do vídeo. Por fim, o vídeo é finalizado sem nenhum objeto presente na cena (Figura 25.d).

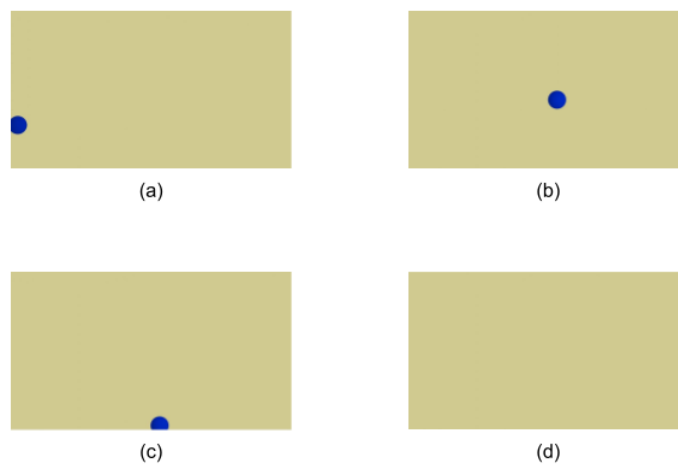


Figura 25 – Quadros do vídeo do Exemplo 2. (a) tempo = 3.12 s; (b) tempo = 5.99 s; (c) tempo = 19.25 s; (d) tempo = 20.79 s.

Fonte – Fonte autoral

Como forma de validar o algoritmo proposto, na Figura 26 são apresentados quadros do vídeo analisado com rastreamento do objeto de interesse. No primeiro quadro (Figura 26.a) o objeto está presente na cena e em vermelho é apresentado o histórico de posições estimadas pelo filtro de Kalman nos frames anteriores. Percebe-se que o rastreamento é realizado corretamente a partir do segundo quadro apresentado (Figura 26.b), tendo em vista a preservação da identidade do objeto de interesse a partir de sua numeração e do seu histórico de posições. No terceiro quadro (Figura 26.c) o objeto está saindo de cena e no último quadro (Figura 26.d) o filtro de Kalman referente ao objeto se enquadra na situação em que há falta de posições medidas. Entretanto, o filtro permanece ativo (por algum tempo) e realizando a estimação a partir da última posição medida e do modelo dinâmico do sistema.

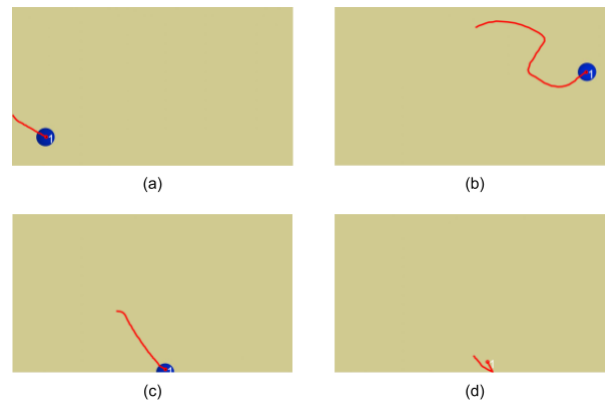


Figura 26 – Quadros do vídeo 2 com rastreamento. (a) tempo = 3.79 s; (b) tempo = 12.79 s; (c) tempo = 19.45 s; (d) tempo = 20.79 s.

Fonte – Fonte autoral

### 4.3 Exemplo 3

O objetivo desse terceiro exemplo é avaliar o sistema de rastreamento em um cenário de que múltiplos objetos (5 no total) aparecem em uma cena. Especificamente, o objetivo é rastrear todos os objetos desde o momento que surgem na cena até o momento que desaparecem. Neste vídeo, conforme apresentado na Figura 27 inicialmente, nenhum objeto está presente na cena e, gradativamente, surgem no lado esquerdo da cena e percorrem seus trajetos horizontalmente até o lado direito. O primeiro quadro (Figura 27.a) apresenta o primeiro objeto de interesse entrando em cena. Na sequência, os outros objetos começam a surgir (Figura 27.b) até que todos são apresentados na cena (Figura 27.c). Finalmente, os objetos começam a sair de cena (Figura 27.d) até que nenhum permaneça visível.

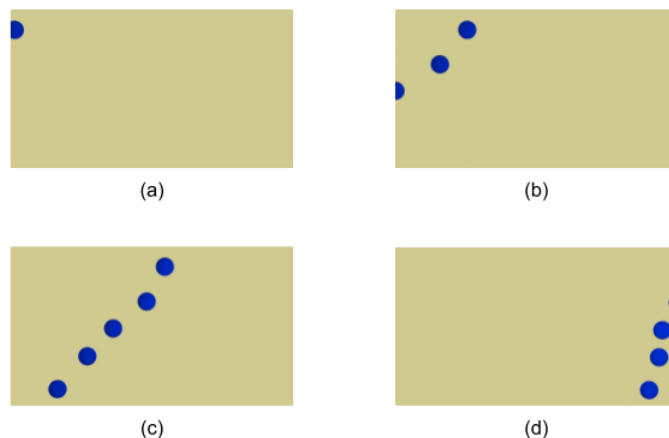


Figura 27 – Quadros do vídeo do Exemplo 3. (a) tempo = 4.58 s; (b) tempo = 7.66 s; (c) tempo = 10.79 s; (d) tempo = 16.29 s

Fonte – Fonte autoral

Na Figura 28 são apresentados quadros do vídeo analisado com o resultado do rastreamento dos objetos. Nota-se inicialmente que três objetos estão presentes em cena e cada um possui um filtro de Kalman com sua respectiva identificação de acordo com o momento em que apareceram na cena (Figura 28.a). No quadro seguinte, todos os objetos de interesse estão visíveis (Figura 28.b) e percorrendo o trajeto horizontal até o lado direito da cena (Figura 28.c). No último quadro apresentado (Figura 28.d) os filtros de Kalman existentes estão submetidos à situação em que as posições medidas dos objetos são desconhecidas. Percebe-se que diferentemente do Exemplo 4.2, alguns filtros já excederam o tempo limite estabelecido pelo projetista (filtros referentes aos objetos 1 e 2) e foram desativados. Entretanto, os filtros que permaneceram ativos realizam a estimação da posição do objeto com base no modelo dinâmico do sistema e na última posição medida.

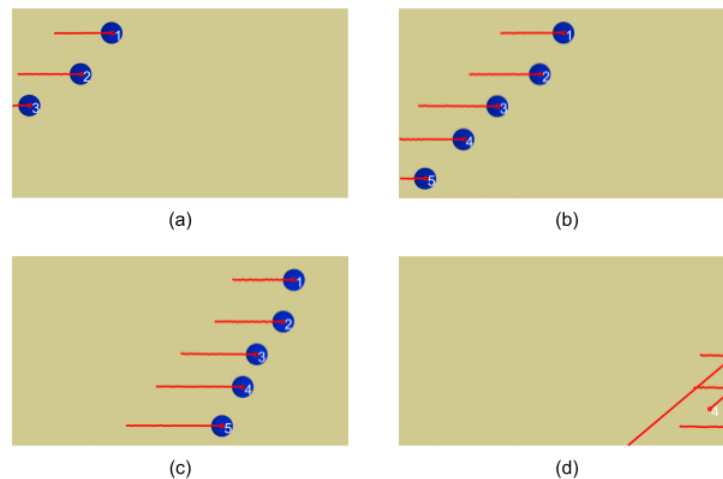


Figura 28 – Quadros do vídeo 3 com rastreamento. (a) tempo = 8.20 s; (b) tempo = 10.25 s; (c) tempo = 13.41 s; (d) tempo = 17.75 s.

Fonte – Fonte autoral

## 4.4 Exemplo 4

O quarto exemplo é composto por dois objetos de interesse que inicialmente não estão presentes na cena, e o objetivo é rastrear as trajetórias de ambos preservando suas identificações. Durante o vídeo os objetos percorrem seus trajetos e saem de cena antes do vídeo ser finalizado. As trajetórias não são uniformes (como no exemplo anterior), sendo que os dois objetos percorrem trajetos que se cruzam ao longo do vídeo. Na Figura 29, inicialmente o primeiro objeto de interesse está presente na cena integralmente enquanto o segundo está começando a aparecer (Figura 29.a). No quadro seguinte (Figura 29.b) os dois objetos estão visíveis e percorrendo seus trajetos. Os quadros inferiores da Figura 29 ilustram, respectivamente, o primeiro e o segundo objeto saindo de cena.

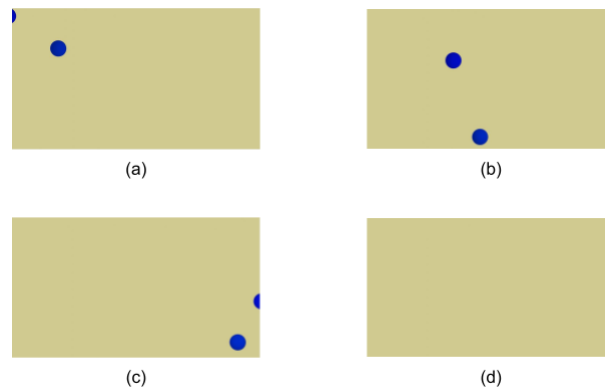


Figura 29 – Quadros do vídeo do Exemplo 4. (a) tempo = 6.04 s; (b) tempo = 9.66 s; (c) tempo = 15.58 s; (d) tempo = 16.54 s.

Fonte – Fonte autoral

Na Figura 30 são mostrados alguns quadros do vídeo com a inclusão do algoritmo de rastreamento. No primeiro quadro (Figura 30.a) apenas um objeto está presente em cena, com a identificação do primeiro filtro de Kalman criado e o histórico das últimas 50 posições estimadas pelo filtro. Posteriormente, os dois objetos de interesse estão presentes na cena com seus respectivos filtros criados e identificados (Figura 30.b). No terceiro quadro (Figura 30.c) as trajetórias dos objetos se cruzam, sendo visíveis a partir dos históricos das posições estimadas pelos filtros. Por fim, no último quadro, nenhum dos objetos está presente na cena (Figura 30.d), entretanto, o filtro referente ao primeiro objeto a sair de cena está desativado, enquanto o do segundo ainda realiza (por algum tempo) a estimação da possível posição do objeto baseada no modelo dinâmico do sistema.

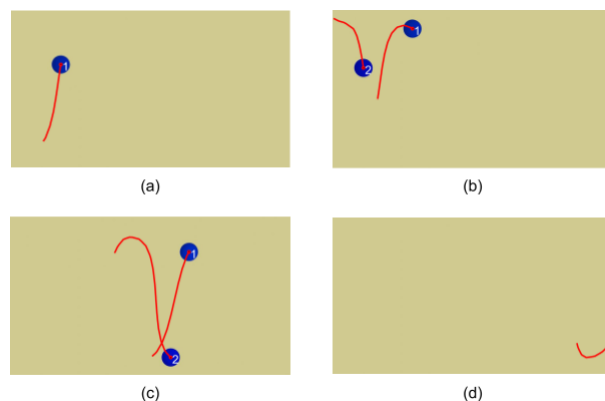


Figura 30 – Quadros do vídeo 4 com rastreamento. (a) tempo = 5.83 s; (b) tempo = 7.16 s; (c) tempo = 12.12 s; (d) tempo = 16.91 s.

Fonte – Fonte autoral

## 4.5 Exemplo 5

O quinto e último exemplo tem como objetivo apresentar o comportamento do sistema de rastreamento na situação em que há uma oclusão momentânea de objetos rastreados. Nesta situação, vale ressaltar que as posições dos objetos não são medidas durante o período em que a oclusão ocorre. Na Figura 31 são apresentados alguns quadros do vídeo analisado. Nota-se que o contexto é similar ao do Exemplo 4.3, onde os objetos percorrem a cena horizontalmente da esquerda para a direita, entretanto, há um objeto retangular verde posicionado verticalmente na região central da cena e está localizado sobre o trajeto dos objetos de interesse. Desta forma, conforme os objetos de interesse passam sob o objeto verde, não é possível obter as informações das posições medidas destes objetos.

No primeiro quadro (Figura 31.a) os objetos de interesse estão entrando em cena, sendo que o primeiro objeto está totalmente visível e o segundo apenas parcialmente. No segundo quadro (Figura 31.b) o primeiro objeto de interesse está oculto pois está sendo sobreposto pelo elemento verde. A mesma situação ocorre para todos os objetos de interesse ao longo do percurso (Figura 31.c). Por fim, no quadro da Figura 31.d, observam-se os últimos três objetos saindo da cena.

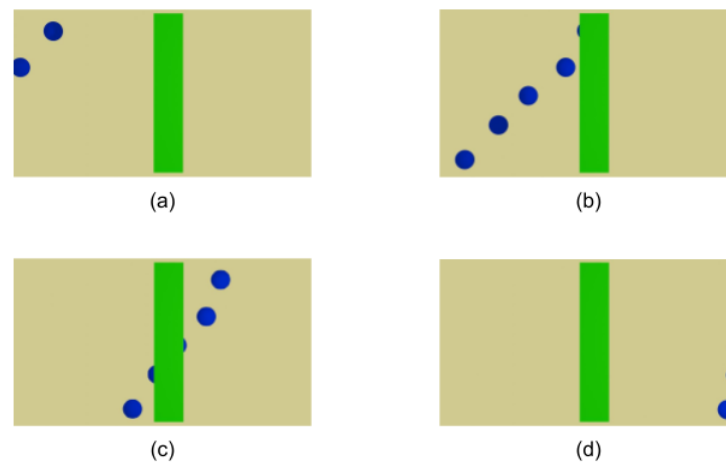


Figura 31 – Quadros do vídeo do Exemplo 5. (a) tempo = 6.20 s; (b) tempo = 10.25 s; (c) tempo = 12.37 s; (d) tempo = 16.95 s.

Fonte – Fonte autoral

Ao aplicar o sistema de rastreamento no vídeo descrito anteriormente, tem-se os resultados apresentados na Figura 32, onde percebe-se que inicialmente os objetos de interesse são identificados corretamente, com o histórico das posições estimadas pelos respectivos filtros de Kalman apresentados na Figura 32.a. No segundo quadro (Figura 32.b), observa-se o efeito da oclusão momentânea dos objetos 1 e 2, nesse caso o filtro de Kalman realiza a estimação da posição dos objetos com base apenas no modelo dinâmico escolhido para representação do objeto. O mesmo ocorre quando os demais objetos de interesse passam sob a região em verde por alguns instantes de tempo (Figura 32.c). Percebe-se que para

o objeto 4 a estimação de sua posição durante o período de oclusão foi determinada precisamente, validando a funcionalidade do modelo dinâmico implementado. Por fim, os objetos de interesse saem de cena, com o último quadro (Figura 32.d) mostrando que os filtros referentes aos objetos 1 e 2 foram desativados enquanto para os objetos 3, 4 e 5 são realizadas ainda algumas estimações de posições com base no modelo dinâmico do sistema.

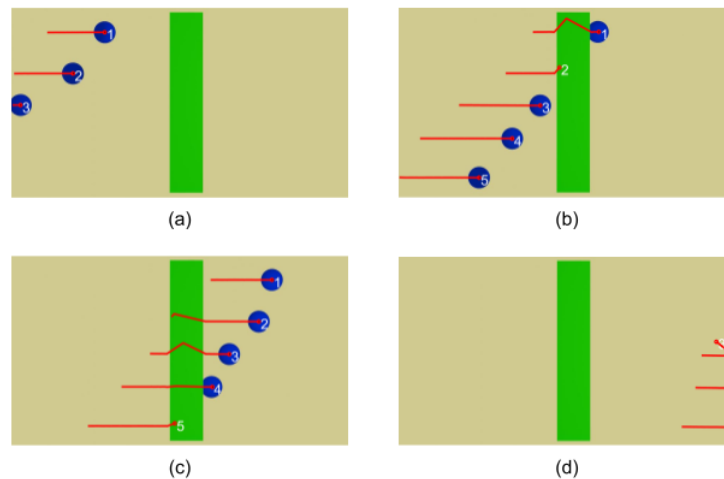


Figura 32 – Quadros do vídeo 5 com rastreamento. (a) tempo = 8.37 s; (b) tempo = 11.12 s; (c) tempo = 13.25 s; (d) tempo = 17.79 s.

Fonte – Fonte autoral

## 4.6 Comentários gerais sobre a complexidade computacional do sistema implementado

Aqui vale destacar que, pelo sistema ter sido implementado no Matlab (que possui uma linguagem de programação interpretada), o tempo de processamento de cada quadro vídeo foi maior do que intervalo de tempo disponível entre quadros consecutivos do vídeo (relacionado com o *frame rate*). Dessa forma, os resultados apresentados nesta seção foram obtidos a partir de um processamento *offline*. Em todo caso, acredita-se que o processamento *online* (em tempo real) possa ser alcançado caso o sistema de rastreamento seja implementado utilizando uma linguagem de programação compilada e com tipagem estática (como, por exemplo, o C++).

Em um computador com processador Intel Core i7-7740X (4.30GHz) e com 12GB de memória RAM, o tempo médio de processamento para cada quadro do vídeo foi de aproximadamente 0,34 segundo, com 0,28 segundo correspondendo a etapa de detecção e localização, e com o restante (0,06 segundo) referente a etapa de rastreamento e gerenciamento dos filtros de Kalman. Para fins de comparação, foram utilizados nos testes

---

vídeos com uma taxa de 24 quadros por segundo (o que corresponde a um quadro a cada 0,0417 segundo). Assim, para que o sistema de rastreamento seja aplicado em tempo real é necessário reduzir o seu tempo de processamento em aproximadamente 8 vezes.

# 5 Conclusões

Neste trabalho foi apresentado o desenvolvimento de um sistema de rastreamento de múltiplos objetos em vídeo utilizando filtro de Kalman. Nesse contexto, no Capítulo 1 foi realizada a introdução ao tema de rastreamento de objetos e as características que o definem. Além disso, foram discutidas as diferenças entre rastreamento e detecção de objetos. No Capítulo 2 foram apresentados alguns conceitos relacionados com a fundamentação teórica desse trabalho. Em especial, foram abordados os conceitos de imagem digital, espaços de cores RGB e CIELAB, alguns algoritmos de processamento digital de imagens (por exemplo, segmentação de imagem, análise de componentes conectados, operações morfológicas e extração de características de região), bem como foi apresentada também a base teórica para a implementação do filtro de Kalman. No Capítulo 3 apresentou-se o sistema de rastreamento de múltiplos objetos desenvolvido neste trabalho. Em particular, foram descritas as duas principais etapas de processamento do sistema, a saber: a detecção de objetos e o gerenciamento dos filtros de Kalman criados para o rastreamento individual de cada objeto de interesse. Finalmente, no Capítulo 4 foram apresentados os resultados da aplicação do sistema desenvolvido em diversos cenários de operação, abrangendo desde situações mais simples (como o rastreamento de um único objeto) até situações mais complexas (como o rastreamento de múltiplos objetos com oclusão momentânea). Foi possível concluir que o sistema proposto obteve sucesso no que diz respeito ao rastreamento dos objetos de interesse presentes em vídeos, na maioria dos cenários avaliados.

## 5.1 Trabalhos futuros

Como forma de aprimoramento do sistema proposto, podem ser implementadas melhorias em relação ao rastreamento dos objetos em situações de oclusões momentâneas, em que a informação da posição do objeto não é fornecida para a etapa de estimação do filtro de Kalman. Neste caso, um modelo dinâmico mais completo (que leve em consideração, por exemplo, a aceleração do objeto rastreado) seria capaz de fornecer informações com maior precisão da posição estimada do objeto no próximo instante de tempo.

Além disso, sugere-se como trabalho futuro a extensão desse sistema para o rastreamento de pessoas em vídeos. O desenvolvimento de tal sistema possui bastante apelo, pois pode ser utilizado em sistemas de segurança e para o controle da quantidade de indivíduos presentes em um local, por exemplo.



# Referências Bibliográficas

- 1 HANXUAN, Y. et al. Recent advances and trends in visual tracking: A review. *Neurocomputing*, Elsevier., v. 74, n. 18, p. 3823–3831, 2011.
- 2 LECUN, Y.; HUANG, F. J.; BOTTOU, L. Learning methods for generic object recognition with invariance to pose and lighting. In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004*. [S.l.]: IEEE, 2004. v. 2, p. 97–104.
- 3 SHANTAIYA, S.; VERMA, K.; MEHTA, K. Multiple object tracking using kalman filter and optical flow. p. 6, 2015.
- 4 KALMAN, R. A new approach to linear filtering and prediction problems. In: *Journal of Basic Engineering*. [S.l.: s.n.], 1960. p. 35– 45.
- 5 JEONG, J.-M.; YOON, T.-S.; PARK, J.-B. Kalman filter based multiple objects detection-tracking algorithm robust to occlusion. In: *2014 Proceedings of the SICE Annual Conference (SICE)*. [S.l.]: IEEE, 2014. p. 941–946.
- 6 CORKE, P. Image formation. In: SPRINGER VERLAG NY. *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. [S.l.], 2011. p. 251.
- 7 CONGALTON, R. G. Exploring and evaluating the consequences of vector-to-raster and raster-to-vector conversion. *Photogrammetric Engineering and Remote Sensing*, [Falls Church, Va.] American Society of Photogrammetry., v. 63, n. 4, p. 425–434, 1997.
- 8 EKEN, S.; SAYAR, A. Performance evaluations of vector-raster satellite image transfers through web services. In: *2012 IEEE 36th Annual Computer Software and Applications Conference*. [S.l.: s.n.], 2012. p. 346–347.
- 9 PEREIRA, S. *Análise estratégica do zoneamento agroecológico como instrumento de ordenamento territorial e sua aplicação em modelos de mudança de uso e cobertura da terra*. Tese (Doutorado), 08 2013.
- 10 ARAÚJO, L. d. L. *Colorização automática de imagens em escala de cinza utilizando Deep Learning*. 2019. Monografia (Bacharelado em Engenharia Elétrica), Universidade Federal de Campina Grande, Paraíba, Brazil.
- 11 SHU, H. et al. Fast computation of tchebichef moments for binary and grayscale images. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, v. 19, p. 3171–80, 12 2010.
- 12 GONZALEZ, R. C.; WOODS, R. E. The rgb color model. In: PRENTICE HALL. *Digital Image Processing*. [S.l.], 2002. p. 290–293.
- 13 GONZALEZ, R. C.; WOODS, R. E. Introduction. In: PRENTICE HALL. *Digital Image Processing*. [S.l.], 2002. p. 1–3.
- 14 PETROU, M.; BOSDOGIANNI, P. Introduction. In: JOHN WILEY & SONS LTD. *Image Processing: The Fundamentals*. [S.l.], 1999. p. 18.

- 15 DUTTA, S.; CHAUDHURI, B. B. A color edge detection algorithm in rgb color space. In: *2009 International Conference on Advances in Recent Technologies in Communication and Computing*. [S.l.: s.n.], 2009. p. 337–340.
- 16 CHANDRAPPA, D. N.; RAVISHANKAR, M.; RAMESHBABU, D. R. Face detection in color images using skin color model algorithm based on skin color information. In: *2011 3rd International Conference on Electronics Computer Technology*. [S.l.: s.n.], 2011. v. 1, p. 254–258.
- 17 MATSUO, M. *LAB 2 - Operações Diádicas e Homografia Planar (Visão Computacional em Robótica)*. [S.l.], 2021.
- 18 Another Perspective. *Activity 9: Morphological Operations*. Disponível em: <<https://adventurousshutterbug.wordpress.com/2017/11/28/activity-9-morphological-operations/>>. Acesso em: 15 setembro 2021.
- 19 THENMOZHI, K.; REDDY, U. S. Image processing techniques for insect shape detection in field crops. In: *2017 International Conference on Inventive Computing and Informatics (ICICI)*. [S.l.: s.n.], 2017. p. 699–704.
- 20 MUNSHI, A. *Morphological image processing operations*. Disponível em: <<https://medium.com/@ami25480/morphological-image-processing-operations-dilation-erosion-opening-and-closing-with-and-without->>. Acesso em: 15 setembro 2021.
- 21 SERRA, J. *Image Analysis and Mathematical Morphology*. [S.l.]: Academic Press, 1982. v. 1.
- 22 GONZALEZ, R. C.; WOODS, R. E. Morphological image processing. In: PRENTICE HALL. *Digital Image Processing*. [S.l.], 2002. p. 528.
- 23 SZELISKI, R. Image processing. In: SPRINGER. *Computer Vision: Algorithms and Applications*. [S.l.], 2002. p. 113.
- 24 WANG, W. et al. A machine vision based automatic optical inspection system for measuring drilling quality of printed circuit boards. *IEEE Access*, v. 5, p. 10817–10833, 2017. ISSN 2169-3536.
- 25 MATSUO, M. *LAB 1 - Análise de Componentes Conectados (Visão Computacional em Robótica)*. [S.l.], 2021.
- 26 CORKE, P. Image feature extraction. In: SPRINGER VERLAG NY. *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. [S.l.], 2011. p. 350 – 355.
- 27 SUN, W. et al. A novel kalman filter based video image processing scheme for two-photon fluorescence microscopy. In: . [s.n.], 2016. v. 9788. Disponível em: <<https://doi.org/10.1117/12.2216129>>.
- 28 SAHBANI, B.; ADIPRAWITA, W. Kalman filter and iterative-hungarian algorithm implementation for low complexity point tracking as part of fast multiple object tracking system. In: *2016 6th International Conference on System Engineering and Technology (ICSET)*. IEEE, 2016. p. 109–115. ISBN 978-1-5090-5089-5. Disponível em: <<http://ieeexplore.ieee.org/document/7849633/>>.

29 MATSUO, M. *Rastreamento de Objetos (Visão Computacional em Robótica)*. [S.l.], 2021.

30 Blender. *Software Blender*. Disponível em: <<https://www.blender.org/>>. Acesso em: 15 setembro 2021.