

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE
AUTOMAÇÃO E SISTEMAS**

Tiago Pinheiro dos Santos

**RASTREAMENTO DE MÚLTIPLOS OBJETOS EM CENAS DE
VIDEOVIGILÂNCIA COM USO DE FILTRO DE KALMAN E
DESCRITORES LOCAIS**

Florianópolis

2018

Tiago Pinheiro dos Santos

**RASTREAMENTO DE MÚLTIPLOS OBJETOS EM CENAS DE
VIDEOVIGILÂNCIA COM USO DE FILTRO DE KALMAN E
DESCRITORES LOCAIS**

Dissertação submetida ao Programa de Pós-Graduação em Engenharia de Automação e Sistemas da Universidade Federal de Santa Catarina para a obtenção do Grau de Mestre em Engenharia de Automação e Sistemas.

Orientador:

Prof. Dr.-Ing Marcelo R. Stemmer

Coorientador:

Prof. Dr. Eng. Jorge H. B. Casagrande

Florianópolis
2018

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Santos, Tiago Pinheiro dos
Rastreamento de múltiplos objeto em cenas de
videovigilância com uso de filtro de Kalman e
descritores locais / Tiago Pinheiro dos Santos ;
orientador, Marcelo Ricardo Stemmer, coorientador,
Jorge Henrique Busatto Casagrande, 2018.
136 p.

Dissertação (mestrado) - Universidade Federal de
Santa Catarina, Centro Tecnológico, Programa de Pós
Graduação em Engenharia de Automação e Sistemas,
Florianópolis, 2018.

Inclui referências.

1. Engenharia de Automação e Sistemas. 2.
Rastreamento visual de múltiplos objetos. 3.
Extração de plano de fundo. 4. Filtro de Kalman. 5.
Descritores locais. I. Stemmer, Marcelo Ricardo.
II. Casagrande, Jorge Henrique Busatto. III.
Universidade Federal de Santa Catarina. Programa de
Pós-Graduação em Engenharia de Automação e Sistemas.
IV. Título.

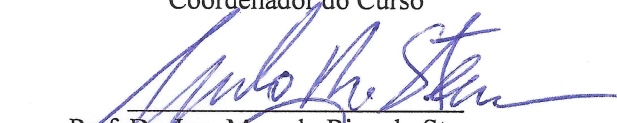
Tiago Pinheiro dos Santos

**RASTREAMENTO DE MÚLTIPLOS OBJETOS EM CENAS DE
VIDEOVIGILÂNCIA COM USO DE FILTRO DE KALMAN E
DESCRITORES LOCAIS**

Esta Dissertação foi julgada adequada para obtenção do Título de
“Mestre em Engenharia de Automação e Sistemas” e aprovada em sua
forma final pelo Programa Pós-Graduação Engenharia de Automação e
Sistemas

Florianópolis, 26 de março de 2018.

Prof. Dr. Daniel Ferreira Coutinho
Coordenador do Curso

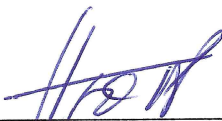


Prof. Dr.-Ing. Marcelo Ricardo Stemmer
Orientador

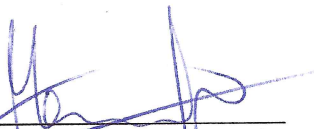


Prof. Dr. Eng. Jorge Henrique Busatto Casagrande
Coorientador

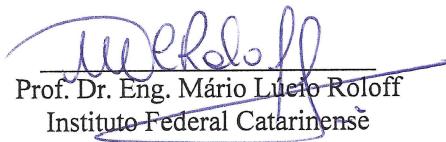
Banca Examinadora:



Prof. Dr. Eng. Tiago Loureiro Figaro da Costa Pinto
Universidade Federal de Santa Catarina



Prof. Dr. Eng. Mauricio Edgar Stivanello
Instituto Federal de Santa Catarina



Prof. Dr. Eng. Mário Lúcio Roloff
Instituto Federal Catarinense

“Não fui eu que lhe ordenei? Seja forte e corajoso! Não se apavore, nem se desanime, pois o Senhor, o seu Deus, estará com você por onde você andar.”

(Josué 1.9)

AGRADECIMENTOS

Gostaria de dedicar essa dissertação a Deus e a todas as pessoas e entidades, como a UFSC e a CAPES, que contribuíram para o desenvolvimento deste trabalho. Quero agradecer a algumas pessoas, em especial, que foram muito importantes para que esse trabalho fosse possível, como meu orientador Prof. Dr.-Ing. Marcelo Ricardo Stemmer e Prof. Dr. Eng. Jorge Henrique Busatto Casagrande, pelos conhecimentos compartilhados; meu pai, Antônio José dos Santos, pelo exemplo de luta e perseverança; minha mãe, Regina Antônia Pinheiro de O. dos Santos, pelo incentivo aos estudos; minhas irmãs Karina e Iris; meu irmão de consideração, Luiz Paulo Enádio dos Reis pelo companheirismo; e minha noiva, Letícia Trindade Almeida, pela motivação extra para concluir mais esta etapa da minha vida.

“...tenha sempre como meta muita força, muita determinação e sempre faça tudo com muito amor e com muita fé em Deus, que um dia você chega lá. De uma forma ou de outra você chega lá.”

Ayrton Senna

RESUMO

Os sistemas de videomonitoramento têm exercido um papel importante na redução da criminalidade, o que torna esses sistemas atrativos para o mercado. O aumento da demanda por sistemas de CFTV (circuitos fechados de televisão) e a necessidade de melhorar a qualidade do trabalho de vigilantes têm impulsionado pesquisadores da área de visão computacional a desenvolver sistemas de videovigilância inteligentes que são capazes de alertar os vigias sobre possíveis ações antissociais no ambiente monitorado. Uma etapa essencial para se criar um sistema desse tipo é o rastreamento de objetos, que é um assunto estudado a mais de 60 anos e ainda motiva pesquisadores a desenvolver métodos e heurísticas criativas para acerrar o problema de rastreamento em aplicações específicas. Este trabalho propõe dois rastreadores de múltiplos alvos voltados para aplicações de videovigilância que utilizam técnicas populares no estado da arte, como a extração de plano de fundo usando o *Visual Background extractor* (ViBe), filtro de Kalman e descritores locais *Fast Retina Keypoint* (FREAK), que são úteis para distinguir um alvo de outro e inferir posições dos mesmos. Esta dissertação aborda os principais tópicos de um rastreador e algumas das soluções encontradas na literatura para este tipo de problema. Os rastreadores propostos apresentaram resultados satisfatórios e competitivos para com alguns algoritmos do estado da arte em sequencias de vídeo específicas.

Palavras-chave: Visão Computacional. Rastreamento de Múltiplos Objetos. Videovigilância. Filtro de Kalman. ViBe. FREAK.

ABSTRACT

Video surveillance systems have been responsible for crime reduction over the years, which is increasing the market appreciation for CCTV (Closed Circuit of Television) systems. The growing demand for those systems and the need to improve the work quality in surveillance have been pushing the scientific community towards intelligent surveillance systems development, which are capable of warning security crew against antisocial behavior at the monitored area. There is a crucial step to build an intelligent surveillance system: the object tracking. Researchers have been studying this topic for more than 60 years, though it still inspires the community to present creative methods and heuristics addressing the object tracking in specific applications. This paper reveals two multiple object tracking algorithms focused on surveillance applications that use popular techniques in the state-of-the-art, such as background subtraction by Visual Background extractor (ViBe), Kalman Filter as an estimator and local descriptors Fast Retina Keypoint (FREAK) to distinguish one target from another, plus it helps target positioning. These algorithms present reasonable and competitive results in comparison with other algorithms in the state-of-the-art considering specific video sequences.

Keywords: Computer Vision. Multiple Object Tracking. Video Surveillance. Kalman Filter. ViBe. FREAK

LISTA DE FIGURAS

Figura 1.1 – Análise de comportamento combinando imagem e áudio ambientes.....	28
Figura 2.1 – Desafios do rastreamento de objetos.....	34
Figura 2.2 – Fluxograma dos principais componentes de um rastreador visual de objetos.....	37
Figura 2.3 – Aplicações de filtros para geração de <i>blobs</i>	41
Figura 2.4 – Representação da propriedade de Markov.....	43
Figura 2.5 – Ilustração do funcionamento do CLD.....	48
Figura 2.6 – Exemplos de textura regular e irregular.....	49
Figura 2.7 – Ilustração de algumas etapas do algoritmo HOG.....	50
Figura 2.8 – Padrão de amostragem BRISK.....	53
Figura 2.9 – (a) analogia entre sistemas biológicos e visão computacional entrada de sinais visuais. (b) concentração de células ganglionares e zonas das retinas.....	54
Figura 2.10 – Padrão de amostragem FREAK.....	54
Figura 2.11 - (esquerda) imagem original de granulados, (centro) imagem convoluída com filtro Gaussiano, (direita) resultado da DoG.....	55
Figura 2.12 – Representação de um descritor SIFT.....	56
Figura 2.13 - Diferença entre SIFT e SURF na construção da pirâmide de escalas.....	57
Figura 2.14 – Fluxograma <i>Mean Shift Tracker</i>	59
Figura 2.15 – Fluxograma de um TLD.....	60
Figura 3.1 – Fluxograma do RAMUK.....	65
Figura 3.2 – Fluxograma do RAMUK-DL.....	66
Figura 3.3 – Representação de valores de <i>pixels</i> em RGB. (a) dois valores dentro da esfera de pertinência e esfera de pertinência; (b) um valor dentro da esfera de pertinência.....	68
Figura 3.4 – Exemplo de dados de saída do RAMUK-DL.....	75
Figura 5.1 – Segmentação ViBe com fantasmas, objetos não detectados e objetos com tamanho errado.....	92
Figura 5.2 – Exemplos de problemas no rastreamento em amarelo.....	92
Figura 5.3 – Detecção ignorada pelo parâmetro de altura mínima.....	93
Figura 5.4 – Fusão de <i>blobs</i> não evitada.....	93
Figura 5.5 – Tratamento de fusão de <i>blobs</i> e de oclusão de objetos.....	93
Figura 5.6 – Associação de pontos-chave entre quadros através do FREAK.....	94
Figura 5.7 – Variação de iluminação repentina e recuperação da detecção.....	94
Figura 5.8 – Média dos modelos de plano de fundo da sequência ShopAssistant2cor.....	98

LISTA DE TABELAS

Tabela 1 – Análise do parâmetro u	87
Tabela 2 – Comparação dos parâmetros ω e σ	88
Tabela 3 – Resultados para o teste de altura mínima	89
Tabela 4 – Resultados para o teste de <i>offset</i>	90
Tabela 5 – Resultados para o teste de pena máxima	90
Tabela 6 – Médias dos resultados de performance para RAMUK e RAMUK-DL.....	91
Tabela 7 – Taxa de processamento dos algoritmos RAMUK e RAMUK-DL sem e com exibição do rastreamento	95
Tabela 8 – Comparativo entre resultados sobre o PETS 2009 - S2L1 ..	96
Tabela 9 – Comparativo entre resultados sobre o CAVIAR 2006	97

LISTA DE ABREVIATURAS E SIGLAS

AGAST – Adaptive and Generic Accelerated Segment Test
BRIEF – Binary Robust Independent Elementary Features
BRISK – Binary Robust Invariant Scalable Keypoints
CAMSHIFT – Continuously Adaptive MeanShift
CAVIAR – Context aware vision using image-based active recognition
CBIR – Content-Based Image Retrieval
CCTV – Closed Circuit of Television
CFTV – Circuito Fechado de Televisão
CLD – Color Layout Descriptor
CUDA – Compute Unified Device Architecture
DAS – Departamento de Automação e Sistemas (UFSC)
DCD – Dominant Color Descriptor
DCT – Transformada Discreta de Cosseno
DMA – Detecção de Movimento Anormal
DoG – Diferença de Gaussianas
EC – Erros de correspondência (troca de rótulos)
EHD – Texture Edge Histogram Descriptor
EKF – Extended Kalman Filter
FAST – Features from Accelerated Segment Test
FK – Filtro de Kalman
FN – Taxa de Falsos Negativos (taxa de perda)
FP – Taxa de Falsos Positivos
FPS – Frames Per Second (quadros por segundo)
FREAK – FAST Retina Keypoint
GLCM – Gray Level Co-occurrence Matrix
GMM – Gaussian Mixture Model
GPS – Global Positioning System
GPU – Graphic Processing Unit
HOG – Histograma de Gradientes Orientados
HSV – Hue Saturation Value (espaço de cores)
INRIA – Institut National de Recherche en Informatique et en Automatique
JPDAF – Joint Probabilistic Data Association Filter
KDE – Kernel Density Estimation
KLT – Kanade-Lucas-Tomasi
kNN – k-Nearest Neighbors
LAP – Linear Assignment Problem
LBP – Local Binary Patterns
LDA – Linear Discriminant Analysis

LoG – Laplaciano de Gaussiana
LPQ – Local Phase Quantization
MILTrack – Multiple Instance Learning Tracker
MHL – Multiple Hypothesis Localisation
MHT – Multiple hypothesis Tracker
ML – Mostly Lost (trilha rastreada por menos de 20% de sua duração)
MOT – Multiple Object Tracking
MOTA – Multiple Object Tracking Accuracy
MOTP – Multiple Object Tracking Precision
MS/MST – Mean Shift Tracker
MT – Mostly Tracked (trilha rastreada por mais de 80% de sua duração)
ORB – Oriented FAST and Rotated BRIEF
PBAS – *Pixel*-Based Adaptive Segmenter
PCA – Principal Component Analysis
PETS – Performance Evaluation of Tracking and Surveillance
PHD – Probabilistic Hypothesis Density Filter
PT – Partially Tracked (trilha rastreada até 80% de sua duração)
RAMUK – Rastreador Multialvos baseado em filtro de Kalman
RAMUK-DL – Rastreador Multialvos baseado em filtro de Kalman e
Descritores Locais
RANSAC – Randon Sample Consensus
RGB – Red, Green and Blue (espaço de cores)
SHL – Single Hypothesis Localisation
SIFT – Scale-Invariant Feature Transform
SURF – Speed Up Robust Features
SVI – Sistema de Videovigilância Inteligente
SVM – Support Vector Machine
TLD – Track Learned Detection
UFSC – Universidade Federal de Santa Catarina
UKF – Unscended Kalman Filter
ViBe – Visual *Background* extractor
VOT – Video Object Tracking Challenge
YCbCr – Luminance, Chroma blue Chroma red (espaço de cores)
YUV – Luminância, matiz e valor (espaço de cores)

SUMÁRIO

1	INTRODUÇÃO	27
1.1	VIDEOVIGILÂNCIA E A VISÃO COMPUTACIONAL ...	27
1.2	SOBRE O RASTREAMENTO DE OBJETOS	29
1.3	OBJETIVOS	30
1.4	CONTRIBUIÇÃO PARA O ESTADO DA ARTE	30
1.5	ORGANIZAÇÃO DA DISSERTAÇÃO	31
2	FUNDAMENTAÇÃO TEÓRICA	33
2.1	DESAFIOS DO RASTREAMENTO DE OBJETOS	33
2.2	CLASSIFICAÇÃO DE RASTREADORES	35
2.2.1	Classificação por grau de interatividade com o usuário..	35
2.2.2	Classificação por estratégias de abordagem	36
2.2.3	Classificação por grau de interatividade com o usuário..	36
2.3	ESTRUTURA E COMPONENTES PRINCIPAIS DE UM RASTREADOR	37
2.4	SEGMENTAÇÃO DE IMAGEM (EXTRAÇÃO DE PLANO DE FUNDO)	38
2.5	EXTRAÇÃO DE CARACTERÍSTICAS	39
2.5.1	Características de nível baixo (<i>low-level features</i>)	40
2.5.2	Características de nível médio (<i>mid-level features</i>)	41
2.5.3	Características de nível alto (<i>high-level features</i>)	42
2.6	LOCALIZAÇÃO	42
2.6.1	Sobre o Filtro de Kalman (FK)	45
2.7	DESCRITORES DE CARACTERÍSTICAS	47
2.7.1	Descritores globais	47
2.7.1.1	Descritores baseados em cores	47
2.7.1.2	Descritores baseados em textura	49
2.7.1.3	Descritores baseados em formas	51

2.7.2	Detecção de pontos de interesse	51
2.7.3	Descritores locais	52
2.7.3.1	<i>Binary Robust Invariant Scalable Keypoints (BRISK)</i>	53
2.7.3.2	<i>Fast Retina Keypoint (FREAK)</i>	53
2.7.3.3	<i>Scale-Invariant Feature Transform (SIFT)</i>	55
2.7.3.4	<i>Speed Up Robust Features (SURF)</i>	56
2.7.4	Fusão de múltiplas características	57
2.8	SOLUÇÕES EM RASTREAMENTO DE OBJETOS	58
2.8.1	<i>Mean Shift Tracker (MST)</i>	58
2.8.2	<i>Continous Adaptative Mean Shift (CAMSHIFT)</i>	59
2.8.3	<i>Multiple Instance Learning (MIL)Tracker (MILTrack)</i>	59
2.8.4	<i>Tracking Learning Detection (TLD)</i>	60
2.8.5	Propostas recentes	60
2.9	OUTROS TRABALHOS RELACIONADOS.....	61
3	IMPLEMENTAÇÃO DOS ALGORITMOS	65
3.1	EXTRAÇÃO DE CARACTERÍSTICAS	67
3.1.1	<i>Visual Background extractor (ViBe)</i>	68
3.2	LOCALIZAÇÃO	71
3.2.1	Importar dados do quadro anterior	71
3.2.2	Implementação do Filtro de Kalman	71
3.2.3	Associação de dados.....	73
3.3	GERENCIAMENTO DE TRILHAS	74
3.4	ARMAZENAMENTO E EXTRAÇÃO DE METADADOS	75
4	MÉTRICAS DE AVALIAÇÃO, DEFINIÇÕES DE PARÂMETROS E PLATAFORMA DE TESTES	77
4.1	MÉTRICAS PARA AVALIAÇÃO DE PERFORMANCE .	77
4.2	MÉTRICAS PARA AVALIAÇÃO DE PERFORMANCE .	80
4.2.1	Parâmetros do ViBe.....	80
4.2.2	Parâmetros do Filtro de Kalman.....	80
4.2.3	Parâmetros da análise de <i>blobs</i>	81

4.2.4	Parâmetros para tratamento de junções de <i>blobs</i>	82
4.2.5	Parâmetros na associação de dados.....	82
4.2.6	Parâmetros para o gerenciamento de trilhas.....	83
4.3	BASE DE DADOS E ANOTAÇÕES DE VÍDEOS (<i>DATASET</i>)	83
5	APRESENTAÇÃO E ANÁLISE DOS RESULTADOS ..	87
5.1	RESULTADOS DOS TESTES DE PARÂMETROS.....	87
5.2	RESULTADOS DE PERFORMANCE	91
5.3	COMPARATIVO COM OUTROS ALGORITMOS NA LITERATURA	95
6	CONSIDERAÇÕES FINAIS	99
6.1	CONCLUSÕES.....	99
6.2	TRABALHOS FUTUROS	100
	REFERÊNCIAS	103
	APÊNDICE I – Resultados do teste de parâmetros para o FK.....	113
	APÊNDICE II - Resultados para o teste de altura mínima	121
	APÊNDICE III - Resultados para o teste de altura mínima	123
	APÊNDICE IV – Resultados para o teste de pena máxima	125
	APÊNDICE V – Resultados de performance do RAMUK e RAMUK- DL	127

1 INTRODUÇÃO

1.1 VIDEOVIGILÂNCIA E A VISÃO COMPUTACIONAL

Adotados como medidas paliativas, os sistemas de videovigilância têm exercido um importante papel na redução da criminalidade, seja inibindo ou prevenindo ações antissociais como infrações de trânsito, violência, vandalismo, roubos e furtos. Essa afirmativa já foi observada há algum tempo: estudos como o de (NIETO, 1997) e (WELSH; FARRINGTON, 2003) comprovam o efeito da implantação de Circuitos Fechados de Televisão (CFTV) na redução da criminalidade, oferecendo mais segurança para pessoas físicas, jurídicas e para o Estado. Apesar de toda polêmica ética e jurídica envolvendo a privacidade das pessoas (KAMPF, 2013; NUNES, 2011; RAJPOOT; JENSEN, 2015), a disseminação desses sistemas tende a aumentar devido à busca por segurança e ao custo acessível dos equipamentos de CFTV.

De acordo com Rätty (2010), Arroyo *et al.* (2015) e Fan *et al.* (2015), a primeira geração dos sistemas de videovigilância (1960-1980) trabalha com sinais de vídeo analógicos, tanto para transmissão quanto para o processamento e armazenamento de imagens. Possui baixo nível de automação e certas restrições quanto a comunicação entre dispositivos. Como a prevenção de crimes ou acidentes está atrelada à atenção constante dos operadores, o que na prática é uma tarefa praticamente impossível, os sistemas da primeira geração, que apenas disponibilizam a imagem para um vigia, ficam vulneráveis devido a falhas humanas. (CASAGRANDE, 2015; RÄTY, 2010). Isso despertou um interesse da comunidade de visão computacional que começou a investigar o uso da inteligência artificial no reconhecimento automático de objetos (RÄTY, 2010).

A segunda geração dos sistemas de videovigilância (1980 – 2000) é marcada pelo uso de sinais digitais e compressão de vídeos, facilitando a transmissão de dados e a comunicação entre equipamentos. Alguns sistemas daquela época já contavam com recursos de auxílio ao vigia capazes de sinalizar eventos relevantes (ARROYO *et al.*, 2015; RÄTY, 2010). Os maiores desafios enfrentados pelos pesquisadores de visão computacional nesse período eram realizar a detecção e rastreamentos de objetos de forma robusta (RÄTY, 2010).

Existe uma divergência entre Rätty (2010), Fan *et al.*(2015) e Arroyo *et al.* (2015) quanto a terceira geração de sistemas de videovigilância. Para Arroyo *et al.* (2015) a terceira geração é

caracterizada por sistemas semiautomáticos capazes de realizar análise de comportamentos, enquanto que, para Rätý (2010) e Fan *et al.* (2015), esse recurso computacional já era explorado na segunda geração. O uso de algoritmos capazes de identificar objetos de interesse, rastrear, classificar e analisar suas trajetórias permitem analisar o comportamento dos alvos. Com isso, surgiram os chamados sistemas de vigilância inteligentes (SVI) como o *IBM Smart Surveillance* (CASAGRANDE, 2015; MAGGIO; CAVALLARO, 2011; YILMAZ; JAVED; SHAH, 2006).

Rätý (2010) e Fan *et al.* (2015) não reconhecem uma quarta geração de sistemas de videovigilância, entretanto as características da quarta geração, denominada por Arroyo *et al.* (2015), aparecem na terceira geração de Rätý (2010) e Fan *et al.* (2015). Os sistemas dessa geração podem monitorar uma grande área, pois eles integram informações de múltiplas câmeras simultaneamente, permitindo o rastreamento de um mesmo objeto por toda a área monitorada. Essa geração usa o conceito de sistemas distribuídos, com algoritmos de processamento de imagens embutidos em equipamentos, permitindo uma análise de dados em tempo real (RÄTY, 2010). Além disso, essa geração combina tipos de sinais diferentes, como por exemplo, sinais de sensores infravermelhos e microfones para auxiliar na análise de eventos. A figura 1 ilustra uma aplicação que combina sinais de diferentes tipos para realizar uma análise de comportamento.



Figura 1.1 – Análise de comportamento combinando imagem e áudio ambientes (Adaptada de Rätý, 2010)

As pesquisas voltadas para essa geração de sistemas envolvem processamento de dados em sistemas distribuídos, fusão de dados multissensoriais, técnicas de monitoramento de múltiplas câmeras e

estudos probabilísticos para se obter uma análise mais compreensiva dos eventos que ocorrem no ambiente monitorado (RÄTY, 2010; ARROYO *et al.*, 2015; FAN *et al.*, 2015).

Em suma, a visão computacional mostra um grande interesse em propor soluções para os SVIs com o intuito de se obter dados mais precisos e, com isso, oferecer um nível maior de segurança a pessoas físicas, jurídicas ou para o Estado.

1.2 SOBRE O RASTREAMENTO DE OBJETOS

O rastreamento visual de objetos é empregado em diversas áreas além da videovigilância, sendo uma etapa fundamental de qualquer sistema automatizado inteligente que utilize processamento de imagens (CASAGRANDE, 2015). As técnicas de rastreamento de objetos podem gerar dados de entrada para inúmeras aplicações como a navegação por GPS (*Global Positioning System*), controle de videogames (KINECT da Microsoft, por exemplo), monitoramento de tráfego de veículos em ruas e estradas, aplicações que envolvem realidade aumentada (SANTOS *et al.*, 2012), obtenção de dados para pesquisas na área da medicina (CHALLA *et al.*, 2011; MAGGIO; CAVALLARO, 2011) ou até mesmo implantadas em um robô que utiliza câmeras para reconhecer e se localizar em um ambiente para iniciar um processo de navegação.

Embora identificar alguns objetos e acompanhar suas trajetórias em uma sequência de imagem seja uma tarefa relativamente fácil para os humanos, o rastreamento robusto de objetos não é uma tarefa trivial para as máquinas. O problema vem sendo investigado pela comunidade científica há mais de 60 anos (CHALLA *et al.*, 2011) e, nos últimos cinco, houve um aumento significativo no número de contribuições sobre esse tema, demonstrando o forte interesse da comunidade científica por essa linha de pesquisa (DUBUISSON; GONZALES, 2016).

Como existem inúmeras aplicações para o rastreamento de objetos, encontrar uma solução que ofereça bons resultados para todas parece ser um grande desafio, pois cada aplicação apresenta características diferentes de outras. Por exemplo, um algoritmo proposto para rastrear jogadores em um jogo de futebol (ARNAUT; COSTA, 2009) tem que atender condições diferentes das enfrentadas por um rastreador de veículos. Portanto, a maioria dos pesquisadores preferem desenvolver algoritmos especializados em determinadas situações para obter bons resultados (DUBUISSON; GONZALES, 2016; MAGGIO; CAVALLARO, 2011).

O potencial de mercado dos sistemas de videovigilância inteligentes e o grande interesse da comunidade de visão computacional em investigar o rastreamento de objetos motivaram o desenvolvimento deste trabalho.

1.3 OBJETIVOS

Objetivo principal: criar a primeira versão de um algoritmo de rastreamentos de múltiplos objetos desenvolvido pelo Departamento de Automação e Sistemas (DAS) da Universidade Federal de Santa Catarina (UFSC), a fim de fortalecer o estudo desta linha de pesquisa no departamento.

Este trabalho também possui os listados objetivos específicos:

- O rastreador deve ser capaz de reconhecer a trajetória de tipos distintos de objetos em imagens oriundas de uma câmera estática de videomonitoramento;
- O rastreamento não deve estar atrelado a um classificação dos objetos;
- Obter resultados competitivos quando comparados com resultados de outros algoritmos da literatura que utilizaram as mesmas base de dados para gerar resultados de desempenho.

1.4 CONTRIBUIÇÃO PARA O ESTADO DA ARTE

O presente trabalho contribui para com o estado da arte através da apresentação de dois algoritmos de rastreamento de múltiplos objetos que seguem a tendência da comunidade científica em utilizar a extração de plano de fundo para a detecção de objetos (ARROYO *et al.*, 2015; V, SPANG, 2014; WU; LIM; YANG, 2015). O primeiro algoritmo realiza o rastreamento associando estados atuais com previsões de estados futuros através de técnicas clássicas como o Filtro de Kalman (FK). O segundo explora a integração do FK com técnicas mais modernas como a correspondência de descritores locais.

Esta dissertação também discorre sobre o uso de técnicas de pós-processamento com filtros morfológicos, detecção de objetos e soluções heurísticas para lidar oclusões de objetos e fusão de aglomerado de *pixels* conectados (*blobs*).

1.5 ORGANIZAÇÃO DA DISSERTAÇÃO

No capítulo 1 é feita uma introdução sobre a importância do videomonitoramento e o papel da visão computacional no desenvolvimento de sistemas de videovigilância inteligentes (SVI's). Este capítulo também discorre sobre o rastreamento de objetos, que é uma importante etapa de sistemas baseados em visão computacional. Ainda neste capítulo são apresentados os objetivos e estrutura desse trabalho. O capítulo 2 discorre sobre tópicos pertinentes ao rastreamento de objetos como os desafios que um rastreador deve superar para seguir um alvo na imagem, a classificação dos rastreadores visuais, detecção de objetos, componentes principais de um rastreador, extração de características, localização dos alvos e associação de dados, além de apresentar as formas de descrever uma imagem ou objeto. O capítulo 3 elucida o desenvolvimento dos algoritmos de rastreamento e as técnicas utilizadas nas implementações. Os parâmetros utilizados, bem como a plataforma de testes desses parâmetros e métodos utilizados para a avaliação do algoritmo são expressos no capítulo 4. No capítulo 5 são apresentados e discutidos os resultados obtidos pelos algoritmos propostos, comparando-os com resultados apresentados por outros trabalhos na literatura considerando a mesma base de dados. O capítulo 6 conclui este trabalho com considerações finais e sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo discorre sobre a base teórica utilizada nesta dissertação, que envolve conceitos relacionados ao rastreamento de objetos e à visão computacional, como detecção de objetos, pós-processamento de imagens, extração de características e técnicas de gerenciamento de trilhas. Também é feita uma análise de alguns trabalhos do estado da arte que se enquadram nesta linha de pesquisa.

O ponto de partida deste trabalho foi a leitura da tese de doutorado de Jorge Henrique Busatto Casagrande (2015), que serviu de motivação para a criação deste projeto. Casagrande (2015) propõe um detector de movimento anormal (DMA) abordando alguns conceitos sobre detecção, descrição e análise de trajetória de objetos. Em sua tese, ele aborda tópicos e cita obras interessantes sobre etapas predecessoras da detecção de anomalias, tais como identificação e rastreamento de objetos, sendo esta última o foco do presente trabalho. Embora Casagrande (2015) não tenha desenvolvido um rastreador em sua contribuição, ele destaca constantemente a importância de se obter dados robustos de um rastreador, pois não é possível analisar comportamentos de objetos de forma satisfatória se as trajetórias dos mesmos se perdem ou são trocadas com a trajetória de outros alvos.

O rastreamento de objetos pode ser entendido como um problema de estimativa das características cinemáticas, tais como posição, velocidade e aceleração (CHALLA *et al.*, 2011; YILMAZ; JAVED; SHAH, 2006; ZHONG; LU; YANG, 2012). De fato, após a detecção dos alvos, uma etapa comumente usada na literatura é estimar essas características cinemáticas, as quais fazem parte dos estados dos objetos de interesse. Todavia, este entendimento omite muitos mecanismos envolvidos para se realizar o rastreamento, os quais serão abordados neste capítulo.

2.1 DESAFIOS DO RASTREAMENTO DE OBJETOS

Os desafios que cercam o rastreamento nem sempre são retratados da mesma forma no estado da arte, pois cada autor aborda o assunto com um grau de detalhamento diferente. Entretanto, Maggio e Cavallaro (2011) apresentam uma figura que resume bem os desafios que um rastreador de objetos deve ser capaz de superar. A figura 2.1 é uma adaptação dessa imagem, na qual foi acrescentado o desafio “fundo dinâmico”.

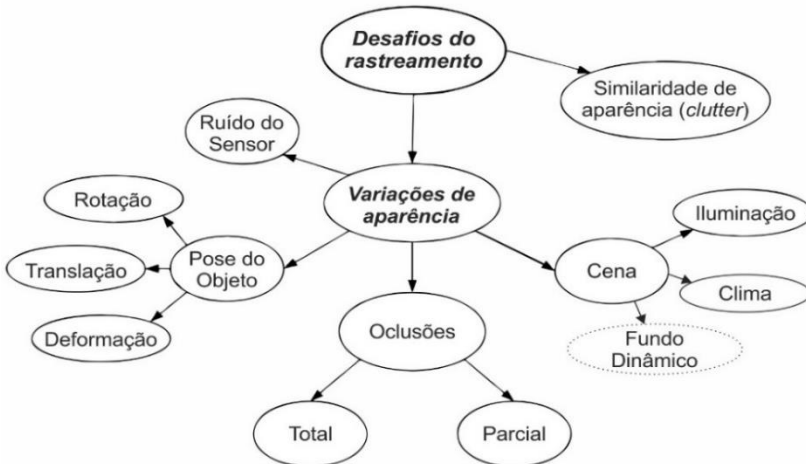


Figura 2.1 – Desafios do rastreamento de objetos
(Adaptada de Maggio e Cavallaro, 2011)

Esta figura está de acordo com os desafios apontados por Casagrande (2015), Challa *et al.* (2011) e Yilmaz *et al.* (2006). Para esses autores, os desafios que tornam complexo o rastreamento robusto de objetos, principalmente quando se trata de identificar um objeto em aplicações para ambientes externos de longa duração (como imagens de videovigilância de ruas e praças), podem ser resumidos em:

- A complexidade que envolvem variações de aparência de objetos não rígidos e plano de fundo não estáticos (dinâmico);
- Ruídos na imagem;
- Variações de iluminação decorrentes de perturbações ou variações climáticas;
- Entrada e saída de objetos no campo de visão;
- Ambiguidades causadas por oclusões dos alvos e similaridade (*clutter*) entre os objetos ou entre objetos e plano de fundo;
- Objetos estáticos por muito tempo podem ser interpretados como *background*;
- Perda de informações na projeção do mundo 3D para uma imagem 2D;
- Requisitos de tempo real (dependendo da aplicação).

Esses desafios geralmente são tratados nas primeiras etapas do rastreamento que é a extração de características. Se não forem levados em consideração na concepção do algoritmo, a performance do rastreador tende a ser pobre, pois não é possível traçar uma boa trajetória de um objeto que não possa ser detectado e identificado corretamente.

2.2 CLASSIFICAÇÃO DE RASTREADORES

A grande variedade de algoritmos de rastreamento na literatura acaba afetando a maneira de como os autores classificam esses algoritmos. Alguns autores como Maggio e Cavallaro (2011) classificam os rastreadores de objetos pelo grau de interatividade com o usuário. Milan *et al.* (2014), Li *et al.* (2013) e Xiang *et al.* (2016) sugerem um agrupamento dos rastreadores de acordo com a estratégia usada para abordar o problema. Já para Casagrande (2015), Xu *et al.*, (2010) e Ko (2008) os algoritmos podem ser organizados com base nas técnicas utilizadas no rastreamento.

2.2.1 Classificação por grau de interatividade com o usuário

Os rastreadores podem ser classificados de acordo com o grau de interatividade com o usuário. Neste caso, eles se agrupam em rastreadores manuais, supervisionados e automáticos.

- *Rastreamento Manual* – são softwares que exigem interações constantes com o usuário para funcionar. Executar o rastreamento nessas plataformas é extremamente trabalhoso, pois requer entradas do usuário praticamente quadro-a-quadro e, portanto, não podem ser usados em aplicações de tempo real. Esses sistemas são preferidos quando a aplicação exige alta precisão e acurácia, como no mercado publicitário audiovisual ou em pós-produção de filmes;
- *Interativos (semiautomáticos)* – essas técnicas são usadas quando se necessita de alguma interação com usuário durante o processo de rastreamento. Por exemplo, um algoritmo que pede para o usuário demarcar uma área na imagem contendo o objeto que se deseja rastrear, como nos trabalhos de Zhong *et al.* (2012), Khan *et al.* (2011);

- *Automáticos* – usam um detector de objetos, como detector de rostos ou extração de plano e fundo, para realizar o rastreamento. Os trabalhos de Choi *et al.* (2015), Roshan Zamir *et al.* (2012) e Berclaz *et al.* (2011) encaixam-se nessa categoria, bem como os algoritmos propostos neste trabalho.

2.2.2 Classificação por estratégias de abordagem

Os algoritmos podem ser classificados de acordo com as estratégias adotadas para resolver o problema de rastreamento. Foram identificados dois tipos de estratégias:

- *Matching-and-tracking* – as soluções que se encontram nesta categoria realizam o rastreamento através da correlação das características dos objetos registradas em quadros anteriores com as características observadas nos quadros atuais. Exemplos de soluções desse tipo são os trabalhos de Nebehay e Pflugfelder (2014) e Maresca e Petrosino (2013);
- *Tracking-by-detection* – de acordo com Pang (2013), este é o tipo de solução mais popular e que atrai mais os pesquisadores. Geralmente a detecção de objetos é baseada em técnicas de segmentação de imagens para separar objetos de interesse e plano de fundo. Li *et al.* (2013) apresenta um levantamento com mais de vinte algoritmos que se enquadram neste tipo. Os algoritmos propostos neste trabalho também se enquadram nessa categoria.

2.2.3 Classificação por grau de interatividade com o usuário

Alguns autores preferem classificar os rastreadores de acordo com as técnicas que fundamentam o funcionamento dos mesmos, principalmente na etapa de associação de dados. Essas categorias são:

- *Rastreamento baseado em características* – podem rastrear um objeto a partir de pontos de interesse utilizando descritores para fazer a correspondência de um ou mais objetos quadro-a-quadro, como por exemplo dos trabalhos de Narayana e Haverkamp (2007), Pang (2013) e (V, SPANG, 2014). Esta dissertação explora também esse tipo de rastreamento;

- *Rastreamento baseado em contorno* – são técnicas que se baseiam em mudanças não significativas nas curvas dos objetos de um frame para outro, ou seja, um objeto é identificado se as diferenças em seus contornos acontecem suavemente de um frame para outro. Para identificar essas mudanças é utilizado o conceito de energia, que identifica bordas, contrastes e cores, mas também pode se fazer o uso de descritores para determinar textura e histograma do contorno. Também é comum o uso de estimadores como filtros de Kalman. Os trabalhos de Yilmaz *et al.* (2004) e Zhang *et al.* (2013) adotam esta abordagem baseada em contorno;
- *Rastreamento baseado em região* – fazem o uso de estimadores para inferir a posição do *blob* (região de *pixels* que representa um objeto). As regiões são obtidas através de dados como energia espaço-temporal, cor, textura, gradiente, ou uma combinação desses dados. Nessa categoria, pode-se citar os trabalhos de Zhou, Yuan e Shi (2009) e Kahn e Gu Backhouse (2011).

2.3 ESTRUTURA E COMPONENTES PRINCIPAIS DE UM RASTREADOR

Mesmo que os algoritmos de rastreamento utilizem técnicas distintas, as soluções apontam alguns elementos em comum para atingir esse fim. A figura 2.2 representa um fluxograma com os principais componentes de um rastreador.

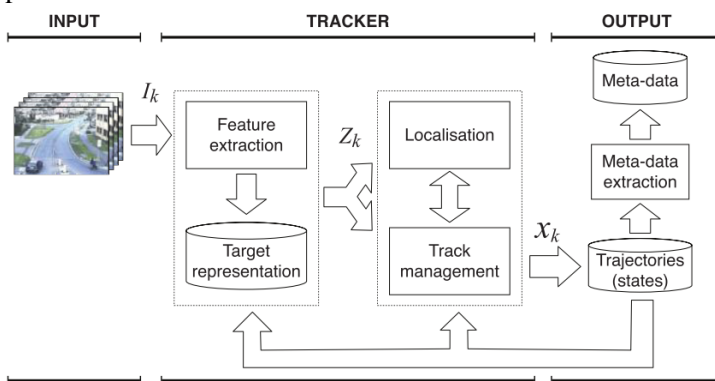


Figura 2.2 – Fluxograma dos principais componentes de um rastreador visual de objetos. (MAGGIO; CAVALLARO, 2011)

Primeiramente, deve-se detectar os objetos e extrair informações relevantes dos alvos como características cinemáticas através de detecções de movimento, classificação de objetos, cor, bordas, etc., e representá-las de alguma forma significativa. Essas informações passam a ser os dados sensoriais do sistema.

Esses dados são utilizados para inferir as posições e, conseqüentemente, as trajetórias dos objetos na imagem atual, geralmente baseados em previsões dos estados em momentos anteriores. Durante a etapa de localização, o algoritmo deve tratar problemas como de oclusão e junção de objetos.

O gerenciamento envolve atividades como inserir ou excluir trilhas, especificar áreas de entrada ou saída dos alvos e atribuir penalidades às trilhas ruins para eventual exclusão das mesmas.

Após o gerenciamento, essas informações são armazenadas em forma de estados dos objetos e são disponibilizadas como saídas do rastreador. Esses estados também são aproveitados pelo próprio algoritmo para continuar rastreando os alvos nas próximas imagens.

2.4 SEGMENTAÇÃO DE IMAGEM (EXTRAÇÃO DE PLANO DE FUNDO)

Dentre todas as soluções propostas para a detecção de objetos, o método de extração de plano de fundo é o mais popular na comunidade da visão computacional (HAN; WANG; CAI, 2016). Neste método, os algoritmos se propõem a modelar o *background* (B) e comparar a imagem atual (I_m) com esse modelo para determinar se um *pixel* da imagem atual $I_m(x,y)$ é pertinente ao plano de fundo ou se esse *pixel* compõe um objeto de interesse.

O estado da arte apresenta distintas formas de se fazer essa comparação, mas de uma maneira geral, elas verificam se a diferença absoluta de intensidades de um *pixel* $I_m(x,y)$ com um *pixel* do modelo de plano de fundo $B(x,y)$ está dentro de um limite (T : *threshold*) aceitável para considerá-lo como um *pixel* de *background*. A expressão (2.1) representa essa comparação.

$$I_m(x,y) \in B \text{ se } |I_m(x,y) - B(x,y)| \leq T \quad (2.1)$$

Parte-se da premissa de que, em aplicações onde o monitoramento é feito por câmera fixa, os *pixels* que permanecem estáticos tem grandes chances de pertencer a um plano de fundo e os que mudam com o tempo poderiam ser considerados como objetos de

interesse. Entretanto, mudanças de intensidade nem sempre devem ser associadas a um alvo, como por exemplo mudanças de iluminação, sombras, reflexos e fundos dinâmicos não deveriam ser consideradas como um objeto de interesse.

Portanto, técnicas muito simples como a detecção de movimentos por diferença entre quadros consecutivos ou uma simples subtração como a representada pela expressão (2.1) não são boas o suficiente para detectar objetos de interesse.

Visando contornar esses problemas, pesquisadores vêm se dedicando à questão da detecção de movimento e propondo soluções interessantes para lidar com os problemas citados, como algoritmos removedores de sombra e segmentadores. Pode-se citar alguns exemplos de segmentadores como o modelo de mistura de gaussianas (GMM), *Pixel-Based Adaptive Segmenter* (PBAS), KDE e variações, W4 e o ViBe – *Visual Background extractor* (CASAGRANDE, 2015; COSTA, 2008; GOYETTE *et al.*, 2012; HARITAOGLU; HARWOOD; DAVIS, 2000; HOFMANN; TIEFENBACHER; RIGOLL, 2012; VAN DROOGENBROECK; PAQUOT, 2012).

As cores podem ser usadas para realizar a segmentação de imagem, entretanto, o aumento da capacidade de discernimento entre plano de fundo e objeto que as cores agregam pode não ser tão impactante quanto o aumento do custo computacional para os algoritmos. Isto é o que foi constatado no estudo de Barnich e Van Droogenbroeck (2011), no qual se comparou o desempenho de alguns segmentadores, inclusive a diferença de desempenho do ViBe em um espaço de cores unidimensional e em um espaços de cores tridimensional; a diferença entre os dois modos de operação é muito pequena, mas a carga computacional é bem maior para o espaço de cores tridimensional.

2.5 EXTRAÇÃO DE CARACTERÍSTICAS

Seguido o raciocínio de Maggio e Cavallaro (2011), para se realizar o rastreamento de objetos é necessário extrair informações úteis sobre os objetos de interesse (*foreground*) ou do plano de fundo (*background*). As características podem ser agrupadas em três níveis: baixo, médio e alto e podem ser usadas tanto no rastreamento baseado em detecção de objetos (*tracking-by-detection*) quanto para o rastreamento baseado em correspondência de pontos (*matching-and-tracking*).

2.5.1 Características de nível baixo (*low-level features*)

Essas características podem ser entendidas como as informações extraídas diretamente dos vídeos e em geral expressam pouca informação semântica sobre a imagem, ou seja, capacidade discriminativa sobre o que é aquela imagem. Pode-se citar como características de baixo nível: cores, invariantes fotométricos, gradientes de imagens e movimentos.

As cores em imagens digitais podem ser representadas de formas distintas dependendo do espaço de cores adotado, sendo o RGB (*Red, Green, Blue*), YCbCr (*Luminance, Chroma blue Chroma red*) e o HSL (HSV) os mais usuais (SILVA, 2016). A cor é uma característica invariante à orientação e escala (SILVA, 2016) e pode contribuir para atingir invariância à mudanças de iluminação (VAN DE SANDE; GEVERS; SNOEK, 2010) mesmo sendo sensíveis a tais mudanças (TAKALA; PIETIKAINEN, 2007).

Invariantes fotométricos são funções que relacionam as cores de um ponto a fim de evitar que variações de iluminação afetem sua descrição. Pode-se citar o RGB normalizado e o H/S (*Hue e Saturation*) como invariantes a mudanças de iluminação até certo ponto (MAGGIO; CAVALLARO, 2011).

Gradientes são robustos às variações de iluminação. Geralmente são utilizados na detecção de contornos e na extração de informações sobre o interior de uma área como textura. O operador Sobel é um gradiente bem conhecido no âmbito da visão computacional e utilizado para detectar bordas. Os gradientes podem ser usados no ajuste da escala de objetos e no tratamento de ruídos (MAGGIO; CAVALLARO, 2011).

Outra característica de baixo nível usada para se detectar objetos de interesse são as características de movimento. Elas podem ser captadas através de vetores que descrevem a variação de intensidade de um *pixel* entre dois quadros (*frame difference*) ou através de análises de fluxo ótico. Características cinemáticas como posição e velocidade podem ser bem úteis no rastreamento de objetos, ou até mesmo fundamentais (GOYETTE *et al.*, 2012). Em caso de uma oclusão de objetos, estas características podem ser usadas para estimar a posição de um objeto, mesmo que ele não tenha sido detectado pela câmera. As características cinemáticas geralmente compõem o estado dos alvos e podem ser usadas com heurísticas para correlacionar objetos quadro-a-quadro.

2.5.2 Características de nível médio (*mid-level features*)

Essas características permitem uma melhor compreensão sobre o contexto de uma imagem. São bastante estudadas pelos pesquisadores e permitem analisar um conjunto seletivo de *pixels* e seus vizinhos com características semelhantes. Esta seleção pode representar uma borda, pontos de interesse ou um aglomerado de *pixels* chamados de região de interesse (MAGGIO; CAVALLARO, 2011).

Em comparação com as cores, as bordas são muito mais robustas às variações de iluminação, pois mesmo que ocorram mudanças desse tipo, a fronteira de um objeto apresenta uma diferença de intensidade relevante que indica o limite de objetos. O detector de borda mais utilizado é o *Canny Edge detector* devido aos seus ótimos resultados (DEORI; THOUNAOJAM, 2014; YILMAZ; JAVED; SHAH, 2006).

Pontos de interesse são pontos que podem ser detectados mesmo quando ocorrem mudanças de iluminação. O detector de Moravec percebe variações de intensidade em *pixels* vizinhos, porém este não é um detector muito usado já que ele sofre com as mudanças de orientação dos objetos. O detector de cantos de Harris já é um método mais adequado para o rastreamento de objetos, pois consegue lidar com mudanças de orientação (MAGGIO; CAVALLARO, 2011).

Em imagens de câmera de segurança, é comum que os alvos mudem sua escala durante com o tempo. Neste caso, ao se aplicar um filtro Gaussiano com diferentes tamanhos de núcleos é possível obter informações dos *pixels* vizinhos de um ponto de interesse. Esse conjunto de *pixels* conectados passam a representar uma região de interesse (MAGGIO; CAVALLARO, 2011). A figura 2.3 ilustra como a convolução de uma imagem com filtros pode ser usada para se gerar *blobs*. A imagem original é convoluída com um filtro Gaussiano, gerando uma imagem mais desfocada. Se ajustar a escala do filtro e aplicar um filtro Laplaciano de Gaussianas, pode-se identificar um *blob*, no qual o centro de massa seria sua posição.

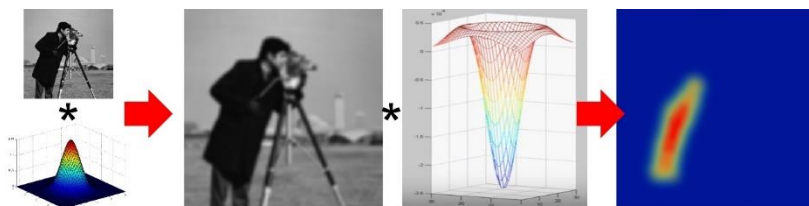


Figura 2.3 – Aplicações de filtros para geração de *blobs*

O SIFT é um descritor de pontos de interesse que faz uso deste artifício para gerar descritores invariantes a escala (MAGGIO; CAVALLARO, 2011).

2.5.3 Características de nível alto (*high-level features*)

Características de nível alto são características que expressam bem o que está sendo mostrado. Por exemplo, um sistema pode usar características para determinar se um conjunto de *pixels* ou área é referente a um cachorro ou uma pessoa. Este é um exemplo de classificação de objetos que pode ou não ser usada no rastreamento de objetos (MAGGIO; CAVALLARO, 2011).

Os algoritmos baseados em modelos de objetos tem a vantagem de poder rastrear objetos em imagens de câmeras em movimento, mas só conseguem rastrear alvos cujo a aparência seja compatível com os modelos aprendidos, necessitando de uma base de treinamento e um classificador de objetos (MAGGIO; CAVALLARO, 2011).

Outra forma de rastrear um objeto envolve a classificação dos *pixels* em modelos de plano de fundo. Algoritmos que fazem essa classificação, também são chamados de segmentadores, agrupam os *pixels* que condizem com os modelos de *background*. Posteriormente, esses pontos são subtraídos da imagem atual e o resultado é uma máscara que contém apenas as regiões dos alvos. Um ponto negativo dessa abordagem é que as imagens que passam por essa segmentação devem ser oriundas de câmeras estáticas (MAGGIO; CAVALLARO, 2011).

O crescente número de rastreadores que se baseiam na extração de plano de fundo tem apontado esse tipo de abordagem como uma tendência na comunidade de visão computacional (ARROYO *et al.*, 2015; V, SPANG, 2014; WU; LIM; YANG, 2015) e isso motivou a escolha dessa abordagem para o desenvolvimento dos algoritmos propostos. Além da extração de plano de fundo, optou-se por extrair características de nível médio e alto para avançar às etapas de localização, gerenciamento de trilhas e extração de metadados.

2.6 LOCALIZAÇÃO

Localizar um ou mais alvos na imagem envolve analisar recursivamente a extração de características dos objetos de interesse, considerando estimativas anteriores para estimar a posição do alvo no frame atual (MAGGIO; CAVALLARO, 2011), sendo que “todos os

algoritmos de rastreamento de objetos utilizam estimadores ou filtros como componente principal” (CHALLA *et al.*, 2011).

Pode-se dizer que o problema de localização se divide em estimar os estados de trilhas existentes no quadro anterior, lidar com problemas de detecção de objetos como oclusões e junção de *blobs* (se aplicável), e associar as estimativas com as detecções observadas.

Segundo Maggio e Cavallaro (2011), existem duas principais classes de métodos de localização: *Single-hypothesis localisation* (SHL) e *Multiple-hypothesis localisation* (MHL). A primeira considera somente uma possibilidade de caminho, ou seja, o algoritmo atribui uma detecção a somente uma trilha possível. A segunda considera múltiplos caminhos como possibilidades, melhorando a performance do rastreador. O processo de associação no MHL é adiado, permitindo que as incertezas geradas durante uma oclusão, por exemplo, sejam sanadas em quadros subsequentes (KIM *et al.*, 2015; MAGGIO; CAVALLARO, 2011).

Alguns exemplos de rastreadores baseados em SHL são o *Mean Shift* (MS) e o Kanade-Lucas-Tomasi (KLT), que são algoritmos de rastreamento baseados em região e usam gradiente de funções para determinar a posição que melhor se assimila com uma região modelo (MAGGIO; CAVALLARO, 2011). Outros rastreadores que trabalham com SHL podem estimar a posição de objetos com base em filtros bayesianos recursivos de posição, como por exemplo o Filtro de Kalman (seção 2.6.1). Esses estimadores possuem a propriedade de Markov (ilustrada pela figura 2.4), a qual alega que a previsão de estado atual depende apenas do estado passado (x_{n-1}) e do estado observado atual (z_n), caso ele exista (CASAGRANDE, 2015; CHALLA *et al.*, 2011; MAGGIO; CAVALLARO, 2011).

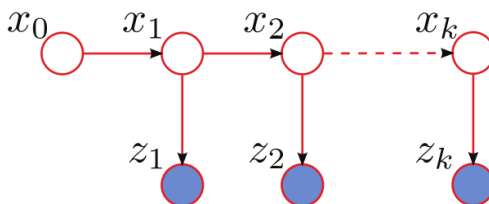


Figura 2.4 – Representação da propriedade de Markov (MAGGIO; CAVALLARO, 2011)

O ponto negativo de algoritmos SHL é que eles dependem de um bom detector de objetos, pois quando a detecção não acontece, a inicialização do estimador de posições fica comprometida, acarretando

em uma estimativas ruins. Portanto, falhas constantes ou prolongadas na detecção geralmente levam o rastreador a perder a trilha do objeto.

Como os métodos MHL armazenam mais de uma trilha possível em cada quadro, caso uma associação tenha sido infortuna, o algoritmo que usa esse método pode se recuperar, mantendo a trilha rastreada sem fragmentá-la. A escolha de qual trilha deve ser mantida e qual deve ser descartada para um objeto é baseada em probabilidades, ou seja, as hipóteses com maior similaridade de características (visuais ou cinemáticas) são propagadas enquanto as demais são descartadas (MAGGIO; CAVALLARO, 2011).

O algoritmo de método MHL mais utilizado no estado da arte é o filtro de partículas, como citado por Maggio e Cavallaro (2011). Este algoritmo é baseado nas técnicas de Monte Carlo e geralmente é adotado quando se deseja estimar estados de sistemas não lineares ou difíceis de modelar (CANNONS, 2008; MILAN; ROTH; SCHINDLER, 2014; OJHA; SAKHARE, 2015; YILMAZ; JAVED; SHAH, 2006). Ele basicamente gera um número de amostras e as propaga segundo uma função. Com os dados de entrada, uma reamostragem é feita na qual os pontos mais prováveis para ser a posição de um objeto tem maior chance de serem propagados. Após algumas iterações o algoritmo converge para pontos bem próximos dos reais.

Embora os métodos MHL têm apresentado melhores resultados com relação à precisão e acurácia com relação aos métodos SHL (MAGGIO; CAVALLARO, 2011), é comum os pesquisadores optarem por soluções SHL pelo seu menor custo computacional e resultados satisfatórios. Considerando essa tendência, o filtro de Kalman foi aproveitado como estimador de estados nos algoritmos propostos.

Prever os estados dos objetos é apenas uma etapa da localização. Ao contrário dos humanos, as máquinas não entendem, a priori, qual detecção é referente a qual objeto. Logo, uma associação de dados é essencial para se criar a trajetória de objetos.

Maggio e Cavallaro (2011) citam duas categorias de soluções para associação de dados: as que se baseiam no conceito do vizinho mais próximo (*nearest neighbor*) e as que procuram uma associação que seja ótima global como um problema de designação linear, também conhecido como *linear assignment problem* (LAP).

No método do vizinho mais próximo, as detecções são associadas às trilhas cujas projeções (previsões) estão mais próximas e este método é o mais simples de se implementar (DEORI; THOUNAOJAM, 2014). Porém, quando os objetos estão muito próximos uns dos outros, as relações entre objetos e detecções podem ser trocadas umas com as

outras. Este problema é conhecido como troca de rótulos (*ID switch*) ou erro de correspondência (*mismatch error*).

Para lidar com esse tipo de erro, a literatura aponta para as soluções MHL, como o MHT, filtros de associação probabilística de dados conjunta (*joint probabilistic data association filter – JPDAF*) e filtros PHD (*probabilistic hypothesis density filter*) (ALBERNAZ, 2016; MAGGIO; CAVALLARO, 2011; MILAN; ROTH; SCHINDLER, 2014; SOUZA, 2015).

Outra solução para o erro de correspondência é combinar métodos baseados em vizinhos próximos com heurísticas para evitar a troca de rótulos. Visando explorar essa possibilidade, utilizou-se o *hungarian algorithm* para buscar o vizinho mais próximo e descritores locais para lidar com a troca de rótulos. Além disso, heurísticas foram adotadas para lidar com a questão da oclusão de objetos e fusão de *blobs*, que são elucidadas no capítulo 3.2.3.

2.6.1 Sobre o Filtro de Kalman (FK)

O Filtro de Kalman é um estimador que considera o estado anterior de um alvo e os dados sensoriais atuais para apontar o estado de um objeto, assim como um filtro bayesiano recursivo. Na verdade, o FK pode ser considerado como um caso específico desse último, pois para que ele seja aplicável, as equações que modelam o sistema devem ser lineares e as variáveis devem seguir uma distribuição gaussiana (CHALLA *et al.*, 2011; MONTANARI, 2015).

Na videovigilância, muitos dos objetos de interesse não seguem um comportamento linear, o que limita o uso do FK para esse tipo de aplicação. Entretanto, autores que empregam o FK nessas situações consideram que, com um tempo de amostragem dos dados dos sensores relativamente curto, pode-se inferir um comportamento linear para os movimentos dos objetos de interesse (CANNONS, 2008).

Além disso, existem outras variações do FK como o *Extended Kalman filter* (EKF), o *Unscented Kalman Filter* (UKF) e outras variações do FK que podem ser aplicadas a sistemas não-lineares (CANNONS, 2008; CHALLA *et al.*, 2011; DEORI; THOUNAOJAM, 2014; MAGGIO; CAVALLARO, 2011).

Algoritmo 1 Filtro de Kalman

1: Previsão de estados e leituras

$$\bar{X}_t = AX_{t-1} + Bu$$

$$\bar{Z}_t = C\bar{X}_t$$

2: Atualizar matriz de covariâncias para cálculo do ganho de Kalman

$$\bar{\Sigma}_t = A\Sigma_{t-1}A' + Q$$

3: Calcular ganho de Kalman

$$K = \bar{\Sigma}_t C' (C\bar{\Sigma}_t C' + R)^{-1}$$

4: Fazer leitura dos sensores em Z_t

$$Z_t \leftarrow \text{dados de sensores}$$

5: Corrigir estimativas

$$X_t = \bar{X}_t + K(Z_t - \bar{Z}_t)$$

6: Atualizar matriz de covariâncias para próxima iteração

$$\Sigma_t = (I - KC)\bar{\Sigma}_t$$

O algoritmo 1 constitui as etapas de um FK. Inicialmente, o algoritmo faz uma previsão do estado de um objeto (\bar{X}_t), tendo como base o estado anterior do alvo (X_{t-1}) e a modelagem matemática que representa o movimento do mesmo (A). A matriz B , que estabelece uma relação entre a ação de controle (u) e o movimento de um objeto, geralmente é omitida por pesquisadores ao se considerar um movimento uniforme para os alvos (CANNONS, 2008).

Em seguida é feita uma previsão de qual seriam os dados de entrada (\bar{Z}_t) oriundos dos observadores considerando a previsão do estado atual e as relações entre dados observados e estados que os produzem (C), ou seja, qual seria a leitura gerada pelo sensor dado certo estado de um objeto.

Posteriormente, a matriz de covariâncias é atualizada para se calcular o ganho de Kalman (K). Após se fazer a leitura dos dados dos observados (Z_t), estes são comparados com a previsão de dados de

entrada (\bar{Z}_t). Essa diferença é multiplicada pelo ganho de Kalman para se fazer uma compensação da estimativa inicial e definir um estado final do alvo (X_t). Por fim, o ganho de Kalman (K), a matriz de correlação entre estado e dados sensoriais (C) e a matriz identidade (I) atualizam a matriz de covariâncias (Σ) que será usada na próxima iteração.

A computação do FK ocorre na forma matricial. Logo, tanto a modelagem do sistema quanto as incertezas sobre o processo e leitura dos sensores devem ser convertidos em matrizes. Essas conversões e parâmetros usados na implementação do algoritmo proposto estão respectivamente discutidos nas seções 3.2.2 e 4.2.2 desta dissertação.

2.7 DESCRITORES DE CARACTERÍSTICAS

Andrade *et al.* (2012) cita uma definição formal de descritores que traduzindo é: “...um par (ϵ_D, δ_D) , onde ϵ_D é um algoritmo de extração para codificação de características visuais (e.g. cor, forma, e textura) em um vetor de características; e δ_D é uma função de similaridade para comparar dados visuais de seus vetores de características.”

Os descritores podem ser classificados em duas categorias: descritores locais e descritores globais (ANDRADE *et al.*, 2012). Os descritores globais ou descritores visuais, segundo (ROYO, 2010), descrevem as características visuais extraídas da imagem/objeto de uma maneira geral (cor, textura, etc.), enquanto que descritores locais descrevem regiões em torno de pontos de interesse da imagem/objeto.

2.7.1 Descritores globais

Devido à simplicidade de implementação e baixo custo computacional, descritores globais são frequentemente utilizados em aplicações de baixo nível como detecção, classificação de objetos e CBIR – recuperação de imagens baseadas em conteúdo (ROYO, 2010). Eles representam objetos ou imagens através de suas características físicas como cor, textura, formas e localização.

2.7.1.1 Descritores baseados em cores

Uma característica muito explorada no estado da arte é a informação de cor dos objetos. Dependendo da aplicação, descrever alvos de uma maneira significativa como uma cor predominante dos mesmos pode ser suficiente para detectá-los na imagem.

Os descritores baseados em cores exigem baixo custo computacional para serem computados e manipulados, além de serem implementados sem muitas complicações. Por essa razão, esses descritores se tornam interessantes para aplicações que envolvam classificação e recuperação de imagens baseada em conteúdo (CBIR), sendo o atributo mais estudado e utilizado em sistemas desse tipo (PENATTI; DA SILVA TORRES, 2008) e em algoritmos de rastreamento (YILMAZ; JAVED; SHAH, 2006).

A cor geralmente é útil para detectar e remover sombras, evitando erros na classificação de *pixels* em objetos de interesse ou plano de fundo e aumenta a robustez de detectores quanto às variações de iluminação (COSTA, 2008; PIVA; SPIRITO; REGAZZONI, 2006; VAN DE SANDE; GEVERS; SNOEK, 2010).

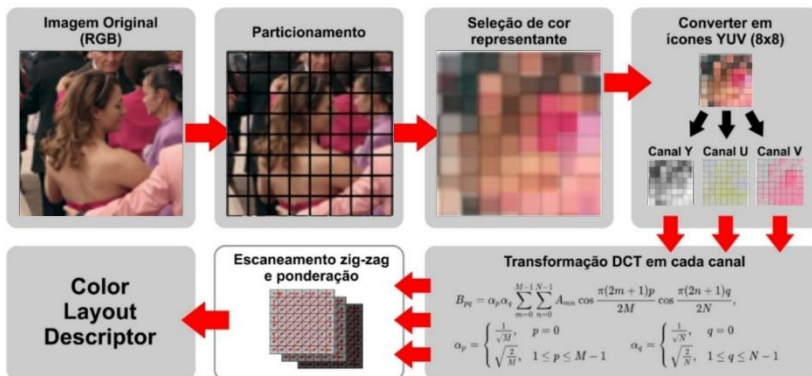


Figura 2.5 – Ilustração do funcionamento do CLD

Um exemplo de um descritor que se baseia em médias de cores é o *Color Layout Descriptor* (CLD). Ele divide a imagem em 64 blocos e descreve as médias das cores de cada bloco no espaço de cores YUV – Y é a luminância enquanto U e V são variáveis de crominância. O poder de síntese desse algoritmo chama a atenção, pois a imagem que outrora era composta por um muitos *pixels* passa a ser representada por um ícone de tamanho 8x8. Os 64 blocos dos 3 canais passam por uma transformada discreta de cosseno (DCT) e seus valores são zigzagueados para formar um vetor de descrição da imagem (ROYO, 2010). O funcionamento do CLD está ilustrado na figura 2.5.

Segundo Royo (2010), existem outros algoritmos baseados em cores como o *Dominant Color* e o *Color Structure* que descrevem uma imagem ou objeto através das cores.

A maior desvantagem desse tipo de descritores é que eles podem encontrar dificuldades para diferenciar um objeto de outro, especialmente em imagens de videovigilância, onde é comum aparecer os objetos pequenos e/ou que possuem cores semelhantes. Por exemplo, roupas escuras são bem comuns e se os objetos de interesse forem pessoas, essa similaridade pode causar confusão entre os objetos de interesse e/ou entre objetos e plano de fundo (*clutter*). Além disso, muitos sistemas de videovigilância contam com fontes de captura monocromáticas e a informação de cor é reduzida para uma dimensão, tornando-se pouco descritiva.

2.7.1.2 Descritores baseados em textura

Uma vez que muitos sistemas de videomonitoramento contam com câmeras monocromáticas, em ordem de se obter um algoritmo de rastreamento mais universal, deve-se utilizar um descritor que possa trabalhar com tons de cinza. Uma solução pode ser o uso de descritores baseados em textura.

Na visão computacional, a textura é percebida através de padrões na região dos objetos como cabelos, vegetação, roupas, etc. As texturas podem possuir repetição de padrões (texturas regulares) ou exibir um padrão aleatório (texturas irregulares), como ilustrado pela figura 2.6. Esses padrões são informações sobre o arranjo estrutural da superfície e sua relação com o ambiente. Foi uma característica bem explorada na última década, principalmente na área de recuperação de informação visual e reconhecimento de padrões (MARTINS *et al.*, 2011; SILVA, 2016).

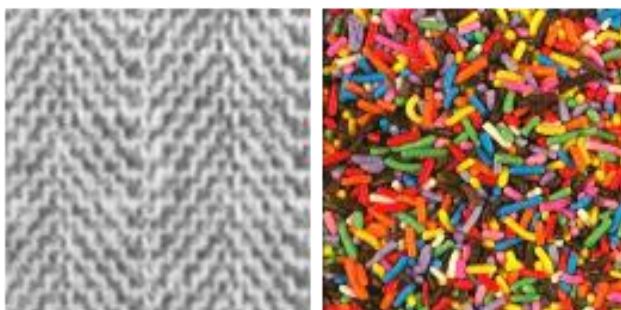


Figura 2.6 – Exemplos de textura regular e irregular (COMUNELLO, [s.d.]

Descritores de texturas geralmente estão atrelados a algoritmos de classificação e suas aplicações variam desde a identificação de espécimes florestais até reconhecimento sinais por espectrogramas (MARTINS *et al.*, 2011). Na videovigilância ele pode ser usado para detectar pessoas e comportamento violento de multidões em imagens de videovigilância (LLOYD *et al.*, 2016).

Os descritores de textura mais comuns na literatura são o *Gray Level Co-occurrence Matrix* (GLCM), *Local Phase Quantization* (LPQ), *Local Binary Patterns* (LBP) e o Histogramas de Gradientes Orientados (HOG).

Se aplicado a algum classificador como SVM, com uma base de treinamento adequada, o descritor HOG pode ser usado para detectar de pessoas (CHOI; MOON; YOO, 2015; DALAL; TRIGGS, 2005). O HOG consiste de uma matriz unidimensional de histogramas, cujo intuito é generalizar a informação de um objeto para que sua representação tenha certo nível de semelhança com a representação de objetos semelhantes, mesmo em diferentes circunstâncias. Por exemplo, para a detecção de humanos, é necessário criar uma representação (vetor descritor) geral o suficiente para que um classificador seja capaz de detectar todas as pessoas em uma imagem, mesmo que elas apresentem diferentes tamanhos (escalas), formas (pose, movimento) e roupas.

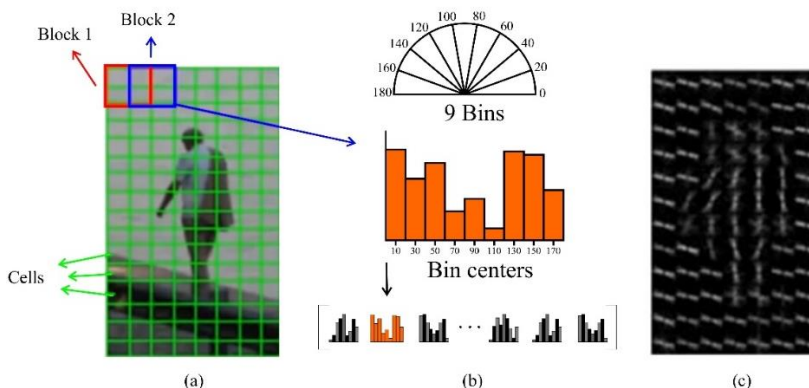


Figura 2.7 – Ilustração de algumas etapas do algoritmo HOG (Adaptado de Shah, 2012)

Na figura 2.7, adaptada de Shah (2012), (a) ilustra uma janela composta por blocos sobrepostos de quatro células com dimensões de 8×8 pixels, então cada bloco possui 16×16 pixels onde 50% deles pertencem também a um bloco vizinho; (b) mostra as faixas de

orientação nas quais os gradientes são classificados e um exemplo do histograma de um bloco que, concatenado com os histogramas de todos os blocos, formam o vetor que descreve o conteúdo da janela de detecção; em (c) pode se visualizar a orientação dominante dos gradientes que coincidem com bordas dos objetos.

2.7.1.3 Descritores baseados em formas

As formas carregam informações importantes para a identidade de objetos (ROYO, 2010). Apenas cor e textura podem não ser determinantes no reconhecimento de objetos, por exemplo, em uma imagem contendo um triângulo e um círculo de mesma cor e textura, a forma pode ser determinante para saber qual objeto é qual.

Um exemplo de um descritor baseado em forma é o *Contour-Shape Descriptor*, que faz a descrição do contorno dos objetos e pode usar atributos como área, perímetro e informações sobre o *bounding box* (ROYO, 2010).

Descritores globais baseados em área e formas podem ser úteis quando aplicados em situações que envolvam objetos rígidos. Objetos dinâmicos e articulados, como pessoas, alteram muito essas características de um quadro para outro, complicando o uso desses descritores para aplicações de videovigilância.

2.7.2 Detecção de pontos de interesse

Antes de abordar a questão dos descritores locais, é preciso conhecer o conceito de pontos de interesse (pontos-chave) ou *keypoints*. Imagine que se deseja montar um quebra-cabeças: ao se olhar para as peças com mesma textura uniforme, como o céu, pode ser difícil de imaginar onde ela será alocada, mas as peças da borda do quebra-cabeças ou que contenham os limites de um objeto podem ser mais fáceis de inferir sua posição.

Seguindo essa analogia, os *pixels* do interior de um objeto totalmente uniforme não nos dizem muito sobre o objeto, pois não é possível diferenciar um ponto do outro pela vizinhança desses *pixels*; é uma região monótona. Entretanto, nas bordas ou cantos de um objeto, ou em texturas irregulares, pela diferença de intensidade dos *pixels* podemos inferir que “algo de interessante” está acontecendo naquela região e que talvez esse ponto pode ser usado para localizar ou delimitar um objeto.

Os cantos de um objeto costumam ser interessantes, e por isso, podem ser usados para gerar dados de entrada para um descritor de características locais. Os *keypoints* são pontos que podem ser identificados em imagens sequenciais, mesmo que um objeto mude sua escala, posição, orientação ou que sofra mudanças de iluminação.

Os primeiros detectores de pontos de interesse foram o detector de cantos de Harris e o detector Shi-Tomassi, que é uma melhoria do algoritmo proposto por Harris. Basicamente, janelas deslizantes são aplicadas ao redor de um ponto para observar as mudanças de intensidades. Ao se usar os gradientes pode-se apontar a curvatura e direção em que as mudanças são mais fortes os pontos onde as mudanças mais fortes ocorrem são apontadas como pontos-chave.

Posteriormente, novos detectores de pontos foram propostos como o Harris-Laplace, DoG (*Difference-of-Gaussians*) e o FAST (*Features from Accelerated Segment Test*) e suas variações (ROSTEN; PORTER; DRUMMOND, 2010).

Alguns descritores como o SIFT (*Scale Invariant Feature Transform*) e BRISK (*Binary Robust Invariant Scalable Keypoints*) incorporam em seu algoritmo uma fase de detecção de pontos-chave. No caso do SIFT, a detecção é feita através da DoG e um algoritmo que identifica máximos e mínimos locais para diferentes escalas e nas vizinhanças de um ponto (LOWE, 1999). O BRISK faz a detecção de pontos-chave baseado no algoritmo AGAST (*Adaptive and Generic Accelerated Segment Test*) que é uma versão melhorada do FAST (ALAHÍ; ORTIZ; VANDERGHEYNST, 2012; LEUTENEGGER; CHLI; SIEGWART, 2011).

2.7.3 Descritores locais

Em muitas técnicas de rastreamento é recorrente o uso de descritores locais, que são algoritmos capazes de extrair características que descrevem as redondezas de pontos de interesse. Como bordas e cantos podem ser identificados em diferentes condições de visibilidade, o uso de descritores locais agregam robustez ao algoritmo de rastreamento que é capaz de identificar um objeto mesmo com variações de iluminação, rotação e escala e oclusões parciais. A seguir discorre-se brevemente sobre alguns dos descritores locais mais comuns na literatura.

2.7.3.1 Binary Robust Invariant Scalable Keypoints (BRISK)

O algoritmo proposto por Leutegger, Chli e Sierwart (2011) realiza a detecção de pontos-chave através de uma solução inspirada no AGAST (MAIR *et al.*, 2010) que é uma variação do FAST. O BRISK escolhe seus *keypoints* considerando uma pontuação FAST máxima que um ponto pode ter e ainda ser considerado como um canto. Os cantos que possuem a pontuação maior do que seus vizinho são mantidos.

A invariância à escala é garantida usando o conceito de pirâmides, onde o fator de escala de cada oitava é 2 e entre oitavas é 1,5 (LEUTENEGGER; CHLI; SIEGWART, 2011; V, SPANG, 2014).

Depois de selecionados os pontos-chave, calcula-se a média ponderada gaussiana ao redor de cada ponto, considerando um padrão de círculos onde os desvios padrões das gaussianas são proporcionais à distância entre os centros destes círculos. A figura 2.8 mostra um padrão de amostragem BRISK com 60 amostras; pontos azuis representam o local das amostras e os círculos vermelhos representam o raio do desvio padrão.

Um descritor binário de 512 bytes é formado ao se comparar essas médias em pares que são escolhido pelo próprio algoritmo (SCHAEFFER, 2013).

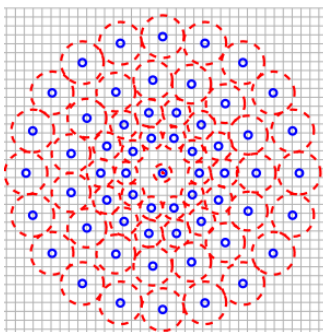


Figura 2.8 – Padrão de amostragem BRISK (LEUTENEGGER; CHLI; SIEGWART, 2011).

2.7.3.2 Fast Retina Keypoint (FREAK)

O principal destaque deste descritor é o fato dele ter sido inspirado no funcionamento biológico de uma retina ocular humana, como observado na figura 2.9.

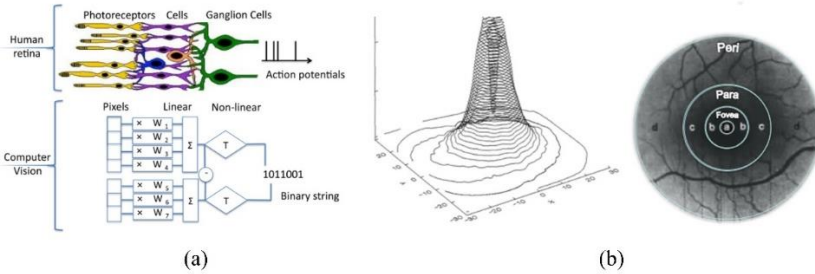


Figura 2.9 – (a) analogia entre sistemas biológicos e visão computacional entrada de sinais visuais. (b) concentração de células ganglionares e zonas das retinas. (ALAHÍ; ORTIZ; VANDERGHEYNST, 2012).

Idealizado por Alahi, Ortiz e Vanderghyest (2012) este descritor melhora o método usado pelo BRISK, pois usa um modelo de amostragem mais descritivo próximo dos pontos-chave. Assim como no olho humano, a definição de uma imagem é maior perto da fóvea devido à maior concentração de pontos de amostragem (células ganglionares). Nesse padrão, representado pela figura 2.10, as Gaussinas se sobrepõem e se concentram mais perto do *keypoint* analisado que pode ser determinado por algoritmos de detecção de pontos-chave como o FAST.

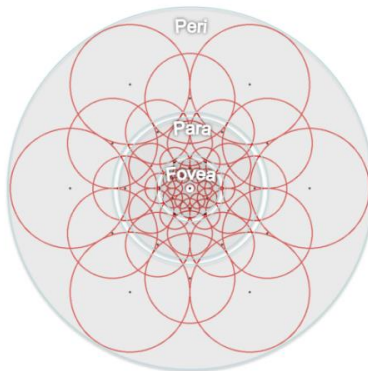


Figura 2.10 – Padrão de amostragem FREAK (ALAHÍ; ORTIZ; VANDERGHEYNST, 2012)

A partir dos pontos-chave, gera-se o vetor binário de descrição F :

$$F = \sum_{0 \leq a \leq N} 2^a T(P_a) \quad (2.2)$$

onde N é o tamanho do descritor, P_a é a intensidade suavizada dos pares r_1 e r_2 , e:

$$T(P_a) = \begin{cases} 1, & \text{se } (I(P_a^{r_1}) - I(P_a^{r_2})) > 0, \\ 0, & \text{caso contrário} \end{cases} \quad (2.3)$$

Os pares são selecionados por um algoritmo que determina os melhores pares não correlacionados para que o vetor de descrição seja mais discriminante (ALAHÍ; ORTIZ; VANDERGHEYNST, 2012).

2.7.3.3 Scale-Invariant Feature Transform (SIFT)

Um dos descritores mais importantes e considerado como referência no estado da arte é o SIFT. Ele faz o uso da diferenças de gaussianas para detectar pontos invariantes a alterações de iluminação, rotação e escala e descrever a vizinhança destes pontos utilizando histogramas de gradientes orientados (LOWE, 1999; V, SPANG, 2014).

Este descritor usa o conceito de pirâmides para obter invariância a escala. Ao se convoluir a imagem com filtros Gaussianos de tamanhos de núcleos diferentes e subtraí-las (DoG), é possível identificar pontos de máxima ou mínima locais; ver figura 2.11.

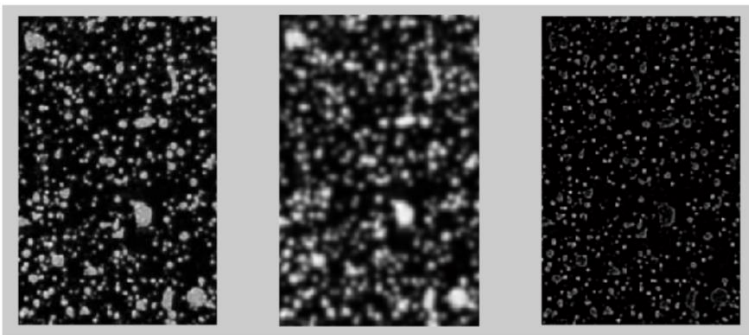


Figura 2.11 - (esquerda) imagem original de granulados, (centro) imagem convoluída com filtro Gaussiano, (direita) resultado da DoG (GONZÁLES, 2010)

Estes candidatos a pontos de interesse são filtrados através de expansão de Taylor e derivadas de primeira e segunda ordem sobre o ponto de extremo. Depois são multiplicados por uma matriz Hessiana 2×2 para eliminar pontos instáveis (GONZÁLES, 2010). Além disso, as imagens convoluídas eliminam ruídos e também tratam pequenas variações de aparência.

Depois de selecionados, os pontos-chave recebem uma orientação para que a descrição desses pontos sejam invariantes à rotação. Ao redor de cada ponto de interesse são calculados os gradientes e magnitudes de 36 *pixels*. Estes histogramas são usados para se formar um descritor de características de 128 valores, conforme na figura 2.12.

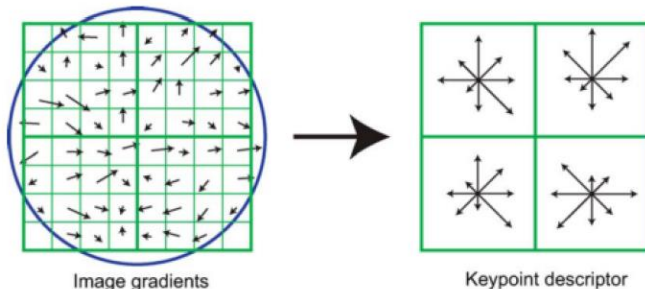


Figura 2.12 – Representação de um descritor SIFT (LOWE, 2004).

Embora o SIFT apresente ótimos resultados de correspondência de pontos, ele é um algoritmo com alto custo computacional, o que limita seu uso em aplicações de tempo real (LEUTENEGGER; CHLI; SIEGWART, 2011; V, SPANG, 2014).

2.7.3.4 Speed Up Robust Features (SURF)

O SURF foi a melhoria mais relevante do SIFT, pois conseguiu manter bons resultados de correspondência com uma velocidade de processamento bem superior ao SIFT (ALAHÍ; ORTIZ; VANDERGHEYNST, 2012; LEUTENEGGER; CHLI; SIEGWART, 2011). Assim como o SIFT, ele gera uma grade orientada, com sub-regiões ao redor dos pontos-chave, calcula o gradiente de cada célula e combina os valores por ângulos para gerar um histograma ponderado por uma Gaussiana.

O SURF faz uso de imagens integrais para convoluções de caixas filtro (*box filter*), que são aproximações dos filtros Gaussianos de derivadas parciais de segunda ordem, com as sub-regiões da imagem. Além disso, ao invés de aplicar filtros gaussianos para obter informações em escala diferentes, as caixas de filtro podem ser aumentadas, o que reduz o custo computacional. Na esquerda da figura 2.13, a imagem mostra a redução da imagem para se aplicar o filtro

Gaussiano, já na imagem à direita, o *box filter* é aumentado para ser convoluído com a imagem.

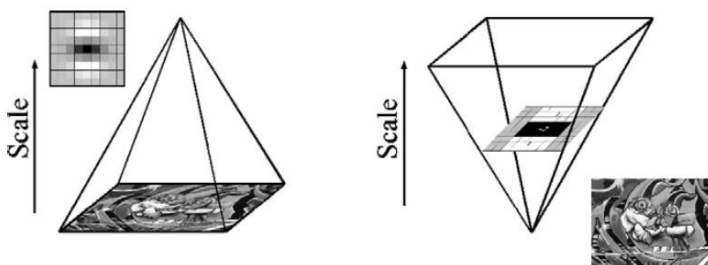


Figura 2.13 - Diferença entre SIFT e SURF na construção da pirâmide de escalas (BAY *et al.*, 2008)

O vetor de descrição do SURF é um HOG de 64 dimensões, mas ele pode ser reduzido, usando PCA (*Principal Component Analysis*), para um vetor de 36 posições. Isso reduz tempo de processamento, permitindo o uso deste descritor em aplicações de tempo real (SCHAEFFER, 2013).

2.7.4 Fusão de múltiplas características

Mesmo a textura podendo ser identificada através dos níveis de brilho da região dos objetos, variações da pose dos objetos podem fazer com que a informação da textura capturada anteriormente seja ocultada (MAGGIO; CAVALLARO, 2011). Por exemplo, se uma pessoa com uma camisa estampada está andando de frente para a câmera e depois ela vira de costas, a informação inicial sobre a textura da camisa é perdida.

Para compensar esse tipo de problema, alguns autores sugerem uma fusão de atributos, que pode integrar características temporais (relacionadas ao movimento), espaciais do objetos (relacionadas as características visuais) e/ou características locais (periferia dos pontos de interesse), gerando um descritor mais completo e com um histórico de melhorar a performance de rastreadores (ANDRADE *et al.*, 2012; DEORI; THOUNAOJAM, 2014; MAGGIO; CAVALLARO, 2011).

A fusão de características pode combinar descritores globais e locais, permitindo aproveitar as vantagens que cada tipo de descritor oferece. Quanto mais informações se puder extrair sobre um objeto, melhores são as chances de identificá-lo.

Entretanto, a quantidade de características a serem combinadas e armazenadas contribui para o aumento no custo computacional. Isto inviabiliza o uso de descritores complexos em aplicações de videovigilância ou em aplicações que apresentam restrições de tempo real. Logo, o número e o tipo de características a serem descritas devem ser adequados para representar um objeto em determinada aplicação. Deve-se ter em mente que podem ocorrer mudanças nas características dos objetos durante o rastreamento, exigindo um mecanismo que gerencie (crie, modifique, exclua) as características rastreadas (MAGGIO; CAVALLARO, 2011).

2.8 SOLUÇÕES EM RASTREAMENTO DE OBJETOS

Nesta seção, são apresentados de forma sucinta alguns dos rastreadores de objetos mais populares no estado da arte, bem como algumas soluções atuais para este problema.

2.8.1 *Mean Shift Tracker* (MST)

O *Mean Shift Tracker*, conhecido as vezes apenas por *Mean Shift* é um rastreador muito usado na literatura para compor soluções de rastreamento de um único objeto. Ele é baseado em *template matching* e foi recentemente adaptado para trabalhar com processamento em GPU (*Graphics Processing Units*) para melhorar o tempo de processamento de imagens (V, SPANG, 2014).

A filosofia por trás desse método é utilizar um modelo de aparência de objetos que se deseja rastrear e, em um próximo quadro, procurar a melhor combinação em uma região vizinha. Ao se variar a região dessa vizinhança (para os lados e o tamanho da região), pode-se criar um vetor que aponta a direção e a magnitude com a qual a região de interesse inicial se moveu. Este processo de busca pela menor diferença entre modelo e janela de busca é repetido até encontrar um mínimo local. Essa nova posição e tamanho do objeto são atualizados para o próximo quadro. Geralmente, a determinação de uma região de interesse inicial é feita de forma manual e a representação do objeto é feita através de histograma de cores (MAGGIO; CAVALLARO, 2011; ZHOU; YUAN; SHI, 2009). A figura 2.14 ilustra o funcionamento do *MST*: a cada quadro, é calculado o histograma de uma região candidata e comparada com o histograma do modelo até a convergência.

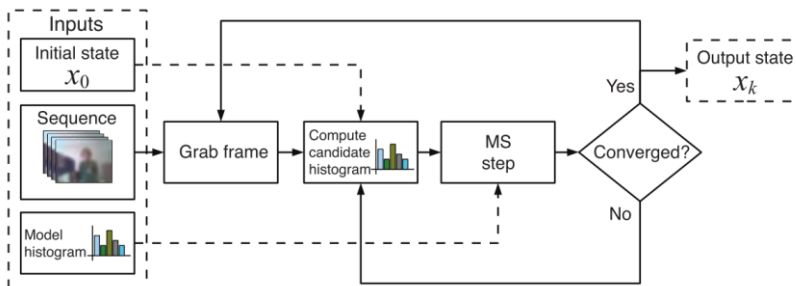


Figura 2.14 – Fluxograma *Mean Shift Tracker* (MAGGIO; CAVALLARO, 2011).

2.8.2 *Continous Adaptive Mean Shift (CAMSHIFT)*

Assim como o *Mean Shift*, este rastreador trabalha com distribuições de probabilidades e seu princípio de funcionamento é encontrar os locais de máxima das modas dessas distribuições. A principal diferença do CAMSHIFT para o MS é que o último trabalha com um tamanho fixo para o objeto, enquanto o primeiro utiliza as saídas do MS e adapta o tamanho e orientação do objeto rastreado (V, SPANG, 2014).

2.8.3 *Multiple Instance Learning (MIL)Tracker (MILTrack)*

Este é um algoritmo também se baseia em *template matching*. Ele foi motivado pelo fato de que incertezas durante a fase de treinamento do modelo de um objeto de interesse podem fazer com que o rastreamento falhe. O MIL trata dessas incertezas acrescentando-as no modelo. Apenas inserindo a informação do centro do objeto, o MIL gera vários retângulos centrados nesse ponto que são armazenados em um conjunto de exemplos positivos levemente desfocados. O rastreador MILTrack também usa exemplos negativos amostrados fora da área do objeto e são selecionados aleatoriamente para compor o conjunto de exemplos negativo. Esses conjuntos de exemplos positivos e negativos são usados para classificar objetos no próximo quadro (V, SPANG, 2014).

2.8.4 Tracking Learning Detection (TLD)

Desenvolvido por Kalal, Mikolajczyk e Matas (2011), TLD é uma sigla que representa as etapas que o algoritmo faz para acompanhar a trajetória de um objeto pela imagem. No rastreamento (*Tracking*), o algoritmo estima o movimento de um alvo e os erros dessa etapa são tratadas pelo detector (*Detection*) em cada quadro. A etapa de treinamento (*Learning*) usa os dados de detecção e o método P-N com o conceito de *Experts P* (detecção de falsos negativos) e *Experts N* (detecção de positivos) para melhorar a detecção de objetos nos próximos quadros. Como são independentes, os experts P e N reduzem os erros um do outro. O TLD também possui um classificador NN (*nearest neighbor*) para classificar partes de uma imagem em exemplos positivos ou negativos (KALAL; MIKOLAJCZYK; MATAS, 2011; V, SPANG, 2014). A figura 2.15 ilustra a interação entre as etapas de um TLD.

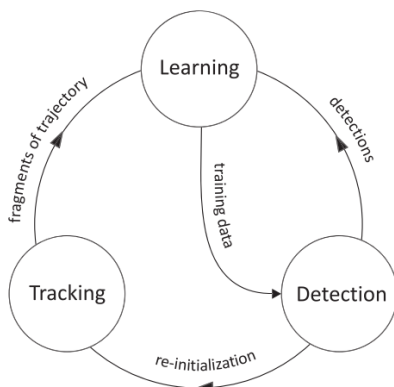


Figura 2.15 – Fluxograma de um TLD (KALAL; MIKOLAJCZYK; MATAS, 2011)

2.8.5 Propostas recentes

Bae e Yoon (2014) apresentam um rastreador de múltiplos objetos baseados em aprendizado de aparências para a detecção de objetos e percentual de confiança nos *tracklets* (segmentos de trilhas) na etapa de associação de dados. O sistema detecta os objetos com base em um conjunto de modelos de aparência que são atualizados conforme o nível de confiança das trilhas. Os resultados do rastreamento foram efetivos inclusive em situações com muitas oclusões e imagens de câmeras móveis.

Choi *et al.* (2015) desenvolveram um rastreador de objetos que utiliza descritores LBP e HOG para detectar humanos em imagens de segurança. Após a subtração do plano de fundo (através de *frame difference accumulation*), a detecção dos objetos de interesse é feita em dois passos: primeiro o LBP gera dados de entrada a um classificador AdaBoost para selecionar as regiões candidatas a possuir humanos; depois são extraídas os descritores HOG, das regiões selecionadas pelo AdaBoost, que entram em um classificador SVM para determinar se aquela região contém ou não um humano. Isso reduz o custo computacional, pois o HOG é executado de maneira objetiva e otimizada (CHOI; MOON; YOO, 2015).

Arroyo *et al.* (2015) propõe um rastreador de objetos que usa descritores de cor (GCH) e de textura (LBP e HOG) para tratar oclusões na etapa de associação de dados, além de usar estimador de posição e velocidade (filtro de Kalman) e uma heurística para resolver linearmente a associação de trajetórias. Foi observado que, em alguns casos, o algoritmo misto obteve resultados superiores à alguns trabalhos recentes no estado da arte (ARROYO *et al.*, 2015).

Elaifi, Jedra e Zahid (2016) apresentam um rastreador de objetos não-supervisionado voltado para o videomonitoramento. O algoritmo utiliza a extração de plano de fundo GMM para a detecção de objetos e o filtro de partículas como motor de inferência de estados. A associação de dados é feita com a ajuda de histograma de cores RGB. Os resultados se mostraram satisfatórios ao se comparar com outros algoritmos na literatura.

2.9 OUTROS TRABALHOS RELACIONADOS

Existem tantas técnicas de rastreamento de objetos no estado da arte que alguns pesquisadores contribuíram com a comunidade científica realizando levantamentos (*surveys e benchmarks*) que agrupam e avaliam o desempenho dessas técnicas. Yilmaz *et al.* (2006) fizeram um levantamento no qual diversos métodos utilizados no rastreamento são classificados com base na representação de objetos e de seus movimentos. O rastreamento pode ser uma tarefa bem complexa e exigir uma grande demanda computacional, as vezes tornando-se inviável para aplicações de tempo real dependendo do método escolhido. Yilmaz *et al.* (2006) discutem formas de reduzir a complexidade do problema, limitando o movimento ou a aparência dos alvos quadro-a-quadro, de acordo com a aplicação que se pretende aplicar o rastreamento. É feito um comparativo detalhado entre os métodos de representação de objetos

e movimentos e se discute características da imagem que ajudam na detecção e diferenciação dos alvos.

No levantamento de Wu, Lim e Yang (2013), feito com o objetivo de medir o desempenho das principais técnicas presentes na literatura, foram avaliados 29 algoritmos de rastreamento, utilizando 50 seqüências de vídeo totalmente anotadas. A robustez de cada algoritmo foi obtida através da análise de mais de 660.000 *bounding boxes*, executando o rastreamento de um único alvo nos vídeos. Os critérios de desempenho tomados foram a robustez, a precisão (distância média euclidiana do centroide obtido para o centroide anotado no *ground truth*) e taxa de sucesso de sobreposição (número de sucesso/número de frames).

Segundo Wu, Lim e Yang (2013), a taxa de sucesso é uma métrica que leva em conta o número de sucesso e é calculada como se segue: é computado um sucesso ($S(f)=1$) quando o valor da divisão da intersecção do número de *pixels* das regiões anotadas (*ra*) e obtidas (*rt*) de um frame pela união do número de *pixels* das mesmas regiões for maior que um limite estipulado entre 0 e 1 (*l*). A equação (2.4) expressa melhor a ideia desse cálculo.

$$\text{Número de Sucesso} = \sum_{f=1}^{\text{total de frames}} S(f); S(f) = \begin{cases} 1, & \text{se } \frac{|rt \cap ra|}{|rt \cup ra|} \geq l \\ 0, & \text{caso contrário} \end{cases}; l = [0; 1] \quad (2.4)$$

Posteriormente, em 2015, os mesmos autores publicaram um novo *benchmark* comparando novos algoritmos e adotando outras medidas de performance. Assim como Spang V (2014) e Arroyo *et al.* (2015), Wu Lim e Yang (2015) apontam uma tendência no estado da arte por abordar o problema de rastreamento através da diferenciação entre objetos de interesse e plano de fundo. Entretanto, assim como no levantamento publicado por eles em 2013, esse comparativo avalia apenas rastreadores de um único objeto (WU; LIM; YANG, 2013, 2015).

Dalal e Triggs (2005) treinaram um classificador SVM com exemplos positivos e negativos de descritores HOG para a presença de pessoas em imagens. Eles obtiveram excelentes resultados na detecção de pessoas, mesmo com diferentes escalas, poses e iluminação, explorando diferentes parâmetros para o algoritmo desenvolvido.

Em uma aplicação de classificação, Martins *et al.* (2011) propôs o uso do GLCM (*Gray Level Co-occurrence Matrix*) que é um método estatístico para descrever a textura de objetos. No trabalho em questão, os autores usaram o GLCM para gerar dados de entrada em classificadores (SVM, LDA, kNN) para diversas finalidades como:

identificação de espécimes florestais por fotos de tronco, reconhecimento de autoria de manuscritos e reconhecimento de gêneros musicais pelos espectrogramas dos sinais (MARTINS *et al.*, 2011). Independente do classificador usado, os resultados obtidos foram muito bons, considerando as bases de dados utilizadas. O GLCM pode ser utilizado também para estimar multidões ou identificar comportamento violento nas mesmas em imagens de videovigilância (LLOYD *et al.*, 2016).

Visando avaliar o desempenho de descritores na classificação de áreas urbanas e rurais em imagens de sensoriamento remoto, Musci *et al.* (2011) fez um comparativo entre o GLCM, LPQ (*Local Phase Quantization*) e LBP, no qual este último obteve os melhores resultados.

Descritores baseados em cores e textura podem não ser suficientes para realizar o rastreamento simultâneo de mais de um objeto, isso porque essas informações podem não diferenciar alvos de cores ou texturas parecidas, como acontece na videovigilância (MAGGIO; CAVALLARO, 2011).

Tendo isso em vista, pesquisadores vêm buscando alternativas para reduzir custos computacionais e realizar a descrição de objetos em imagens de segurança. Uma das vertentes exploradas para atingir esse objetivo é o uso de descritores binários. Pang *et al.* (2013) publicou um artigo sobre um rastreador de objetos robusto que reduz a influência de problemas conhecidos no rastreamento como iluminação de cena, oclusão de objetos, distorção de perspectivas, entre outros. Ele utiliza um classificador binário baseado em BRIEF chamado de ORB que se mostrou invariante às rotações dos objetos e ruídos na imagem.

Ainda sobre descritores binários, Spang V (2014) discute em sua dissertação a variedade de descritores de pontos de interesse que podem ser aplicados em um rastreador de objetos genérico. Spang V (2014) compara os descritores SIFT, BRIEF, ORB, BRISK e FREAK, bem como os rastreadores de objetos consolidados na literatura como o MILTrack, TLD e CAMSHIFT. Spang V (2014) também propõe um algoritmo de rastreamento baseado em descritores locais. De acordo com os testes realizados em cenários distintos, o descritor binário que apresentou melhor acurácia e robustez para o algoritmo desenvolvido foi o FREAK, mas o tempo de processamento deste descritor foi o maior.

Descritores que trabalham com fusão de atributos vêm ganhando o interesse de pesquisadores da área de visão computacional, porque geralmente apresentam melhores performances (ANDRADE *et al.*, 2012; ARROYO *et al.*, 2015; ROYO, 2010). Ilea e Whelan (2011) fizeram uma revisão no estado da arte sobre descritores que juntam os atributos cor e textura. Essas duas importantes características físicas

para descrição de objetos geralmente são utilizadas em concomitância para compensar o efeito das mudanças de iluminação, obtendo um descritor mais completo. Essa junção de atributos tem sido um dos tópicos mais investigados no estado da arte (ILEA; WHELAN, 2011).

Seguindo essa linha de fusão de dados, Royo (2010) propôs um descritor híbrido que concatena os descritores DCD, CLD e EHD. Ele comparou a performance desse novo descritor com os resultados dos descritores implementados individualmente e concluiu que o descritor híbrido apresentou melhores resultados. Embora o estudo tenha sido direcionado para a recuperação de imagens, se adaptado para seguir um modelo de aparência, a proposta de Royo (2010) pode ser usada no rastreamento de alvos.

Segundo Andrade *et al.* (2012), a maioria dos esforços da comunidade científica não explora a junção de descritores de tipos diferentes. Com o intuito de investigar o poder dessa fusão de atributos em problemas CBIR, Andrade *et al.* (2012) apresentou um estudo que comparou o desempenho de descritores locais e globais “puros”, combinados com descritores do mesmo tipo e combinados com descritores de tipos diferentes. Foram avaliados os descritores globais baseados em cor (ACC, BIC, GCH e JAC), baseados em textura (LAS e QCCH), além dos descritores locais SIFT e SURF. Segundo os autores, os algoritmos com fusão de dados apresentaram melhor performance na recuperação de imagens.

3 IMPLEMENTAÇÃO DOS ALGORITMOS

Este capítulo apresenta os fluxogramas dos algoritmos propostos, discutindo as técnicas implementadas em cada etapa e o processo de tomada de decisão por elas, quando aplicável. Os algoritmos foram implementados em MATLAB®, assim como algumas funções no código que estão disponíveis no Computer Vision System Toolbox de vídeo processamento da própria MATHWORKS®.

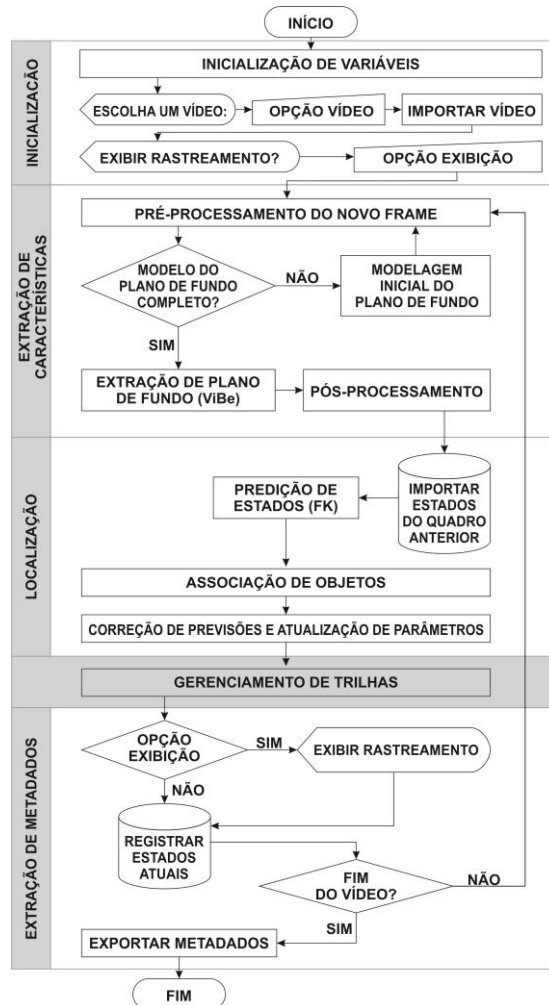


Figura 3.1 – Fluxograma do RAMUK

O RAMUK (Rastreador Multialvos baseado em Filtro de Kalman), cujo fluxograma está representado na figura 3.1, é um algoritmo desenvolvido para usar somente o filtro de Kalman na etapa de localização.

O RAMUK-DL (Rastreador Multialvos baseado em Filtro de Kalman e Descritores Locais), representado pelo fluxograma da figura 3.2, é uma versão mais complexa do RAMUK, pois além do FK, são utilizados descritores locais para auxiliar a etapa de associação de dados.

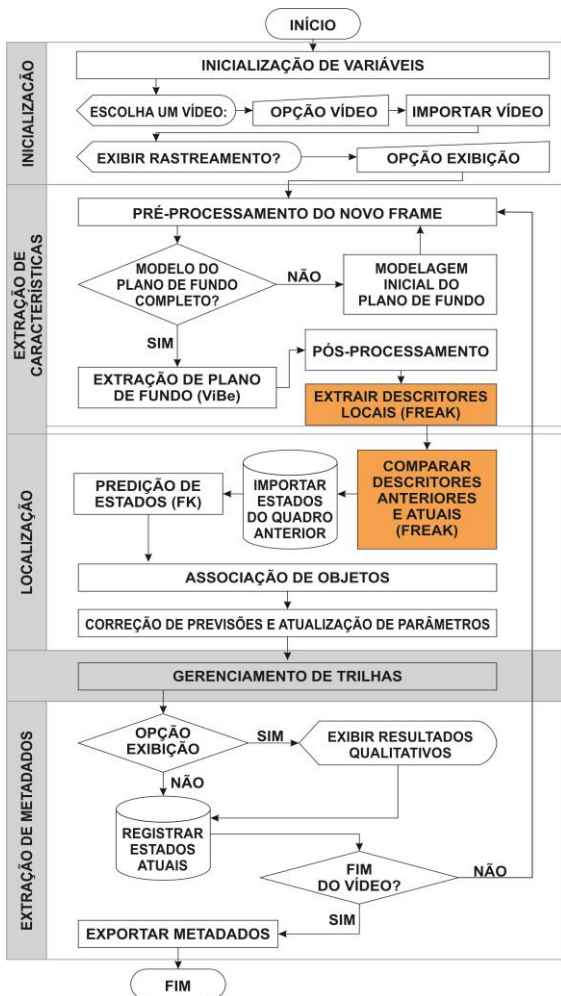


Figura 3.2 – Fluxograma do RAMUK-DL. Quadros em laranja destacam as principais diferenças entre RAMUK e RAMUK-DL.

Como a diferença entre o RAMUK e o RAMUK-DL é praticamente o uso dos descritores na associação de dados, o funcionamento deles será discutido como se fossem o mesmo algoritmo e as diferenças entre eles salientadas nos momentos oportunos. Os fluxogramas dos algoritmos desenvolvidos nesta dissertação, foram formulados com base na divisão de componentes sugerida por Maggio e Cavallaro (2011).

3.1 EXTRAÇÃO DE CARACTERÍSTICAS

Após a inicialização de variáveis e parâmetros usados pelo rastreador, o primeiro componente se inicia. A extração de características foi composta por 4 etapas: pré-processamento, modelagem inicial do plano de fundo, extração de plano de fundo e pós-processamento.

No pré-processamento é feita a leitura do frame atual reduzindo o espaço de escala RGB para tons de cinza. Em seguida aplica-se um filtro passa baixa Gaussiano de tamanho 10x10 com desvio padrão de 5 *pixels*, para remover ruídos e amenizar pequenas variações de aparência do *background*.

Se o número de modelos do plano de fundo não tiver sido atendido, repete-se o passo de montagem do modelo que consiste em receber frames aleatórios pré-processados e os insere com exemplos de plano de fundo.

A escolha do método para a detecção de objetos foi tomada levando em consideração a preferência da comunidade de visão computacional pelos métodos de segmentação de imagens para rastrear objetos em tempo real. Para determinar qual algoritmo seria adotado, considerou-se a disponibilidade de informações que permitissem a reprodução fidedigna do mesmo em MATLAB[®]. Também foi levado em consideração o estudo comparativo de Goyette *et al.* (2012) que apontou os melhores de resultados entre algoritmos de segmentação sob um conjunto de vídeos pertinentes com imagens de câmeras de segurança. Dessa forma, o ViBe (VAN DROOGENBROECK; PAQUOT, 2012) foi selecionado para a detecção de objetos.

A saída do ViBe, mais precisamente a matriz de *pixels* referente àqueles de *foreground*, passa por um pós-processamento para eliminar ruídos e melhorar a extração. O pós-processamento aplica filtros morfológicos (erosão e dilatação) sobre a matriz de saída para eliminar detecções de pequenos grupos de *pixels* isolados e preencher buracos dentro de possíveis alvos. O resultado é uma matriz (máscara) binária que representa os *pixels* de possíveis objetos de interesse. Também são extraídas informações sobre os *blobs*, como área, posição dos centroides

e dimensões dos *bounding boxes* através da função *regionprops()* do MATLAB®.

O RAMUK-DL ainda possui uma etapa a mais na extração de características: a máscara binária é aplicada sobre a imagem para otimizar o processo de detecção de pontos-chave usando o algoritmo FAST e a geração de vetores de descrição FREAK, ambos disponibilizados pelo Computer Vision System Toolbox do MATLAB®.

As informações extraídas dos candidatos são armazenadas em uma variável estruturada temporária, que será usada para a associação de dados no componente de localização.

3.1.1 Visual Background extractor (ViBe)

Desenvolvido inicialmente por Barnich e Van Droogenbroeck (2009), o *Visual Background extractor* é um algoritmo baseado em *pixels* para modelar um plano de fundo e classificar pontos da imagem como pertinentes ao *background* ou ao *foreground*.

O modelo de plano de fundo ($M_n(x, y)$) gerado pelo algoritmo é composto por n amostras e é atualizado de forma aleatória conforme a classificação dos *pixels* da imagem atual. A classificação dos *pixels* se dá através do número mínimo (c) de modelos do *background* que se encontram dentro de uma esfera de pertinência $S_R(p_t(x, y))$ de raio R , conforme ilustrado na figura 3.3.

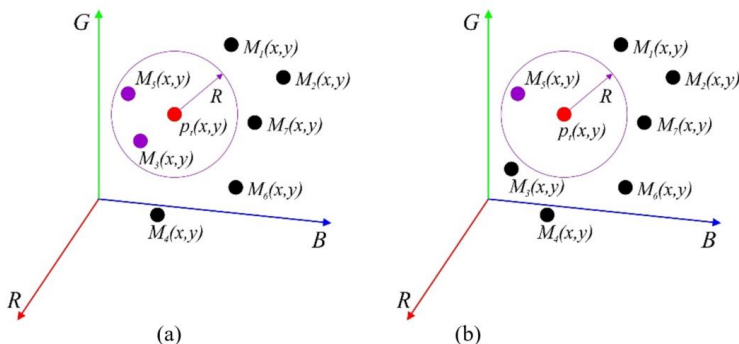


Figura 3.3 – Representação de valores de *pixels* em RGB. (a) dois valores dentro da esfera de pertinência e esfera de pertinência; (b) um valor dentro da esfera de pertinência.

Como ilustrado na figura 3.3, os valores dos *pixels* podem ser representados em um espaço de cores tridimensional, como RGB, HSV,

YCbCr, entre outros. Logo, o termo “esfera de pertinência” pode ser entendido como proximidade de intensidades um *pixel* da imagem atual e os *pixels* do modelo $M_n(x, y)$. Em outras palavras, se o valor de $p_t(x, y)$ for próximo dos valores de ao menos c modelos, o *pixel* é considerado como de plano de fundo. Se $c = 2$, então o *pixel* $p_t(x, y)$ na figura 3.3 (a) será considerado como *background* enquanto o *pixel* na figura 3.3 (b) é classificado como *foreground*.

Se a imagem for em escala de cinza, então o raio de pertinência R passa a ser um limite de similaridade. Dessa forma, a pertinência de um *pixel* ao plano de fundo pode ser definida usando a diferença absoluta entre as intensidades dos *pixels*, através das expressões (3.1) e (3.2).

$$\sum_n P_n = \begin{cases} 1, & \text{se } |p_t(x, y) - M_n(x, y)| \leq R \\ 0, & \text{caso contrário} \end{cases} \quad (3.1)$$

$$p_t(x, y) = \begin{cases} \in \text{background}, & \text{se } P_n \geq c \\ \in \text{foreground}, & \text{caso contrário} \end{cases} \quad (3.2)$$

O algoritmo 2, adaptado de Barnich e Van Droogenbroeck (2011), mostra o passa-a-passa do ViBe. A saída do algoritmo (*FG*) é uma matriz binária onde “1” representa *pixels* de *foreground* e “0” *pixels* de *background*.

Algoritmo 2 Visual Background Extractor - ViBe

- 1: **Início:** Definição de parâmetros
- 2: $N = 20$; $R = 20$; $c = 2$; $\emptyset = 16$
- 3: Criar modelo de plano de fundo inicial
- 4: **Entrada:** $I_t \leftarrow$ frame atual t
- 5: **Para** todos os *pixels* de um frame t
- 6: $p_t(x, y) \leftarrow I_t(x, y)$
- 7: $n \leftarrow 1$
- 8: $P \leftarrow 0$
- 9: **Enquanto** $n \leq N$ e $P < c$, **faça**
- 10: Calcular a distância euclidiana entre $p_t(x, y)$ e $M_n(x, y)$
- 11: **Se** Distância Euclidiana $\leq R$, **faça**

```

12:       $P \leftarrow P + 1$ 
13:      Fim Se
14:       $n \leftarrow n + 1$ 
15:      Fim Enquanto
16:      Se  $P \geq c$ , faça
17:          Armazena pixel como background:
               $FG_t(x, y) \leftarrow 0$ 
18:           $k \leftarrow \text{aleatório}(1; 16)$ 
19:          Se  $k = 1$ , faça
20:              Atualiza um modelo de plano de fundo qualquer do pixel atual:
                   $M_{\dots}(x, y) \leftarrow p_t(x, y)$ 
21:              Fim Se
22:               $k \leftarrow \text{aleatório}(1; 16)$ 
23:              Se  $k = 1$ , faça
24:                  Atualiza um modelo qualquer de um pixel vizinho:
                       $M_{\dots}(\text{viz}(x), \text{viz}(y)) \leftarrow p_t(x, y)$ 
25:                  Fim Se
26:              Senão
27:                  Armazena pixel como foreground
                       $FG_t(x, y) \leftarrow 1$ 
28:              Fim Se
29:      Fim Para
31: Saída:  $FG_t$ 
32: Repetir passos de 4 a 32 até fim do vídeo

```

O ViBe é capaz de começar a segmentação a partir do segundo frame sem a necessidade de treinamento. Para isso, no seu processo de inicialização, assume-se que os valores de *pixels* para frames subsequentes

sejam próximos. Logo, a matriz de modelos $M_n(x, y)$ é preenchida com valores aleatórios na vizinhança de um *pixel* (KRYJAK; GORGON, 2013).

Outra característica marcante do ViBe é a natureza aleatória de amostragens para atualizar o modelo de plano de fundo, gerando resultados de segmentação diferentes para um mesmo vídeo a cada vez que o algoritmo é executado.

3.2 LOCALIZAÇÃO

O componente de localização é formado por 3 etapas principais: Importação de dados do quadro anterior, predição de estados por FK e associação de dados. O RAMUK-DL também possui uma etapa extra neste componente: a comparação dos vetores de descrição dos estados passados armazenados e dos candidatos atuais. Esta é uma etapa chamada de *matching descriptors* e também está disponível no Toolbox do MATLAB®. Após a associação de dados, os algoritmos corrigem as previsões e atualizam os parâmetros do FK para a próxima iteração.

3.2.1 Importar dados do quadro anterior

O primeiro passo é resgatar os estados dos objetos em quadros anteriores. Para o RAMUK, resgata-se somente informações do último quadro como: *status* (se a trilha está ativa ou não), posição e velocidade do objeto, altura e largura do *bounding box*, área do *blob* e penalidade (quantidade de pontos negativos que aquela trilha já recebeu). No RAMUK-DL ainda são resgatadas informações sobre os descritores armazenados sobre o objeto daquela trilha.

3.2.2 Implementação do Filtro de Kalman

A aplicação do filtro de Kalman é bem comum no rastreamento, mas como esse filtro exige um comportamento linear do sistema para funcionar de maneira adequada, o tempo de amostragem (dt) deve ser suficientemente curto para que o movimento dos objetos seja modelado por um sistema linear e, geralmente, assume-se movimentos uniformes para os objetos (CANNONS, 2008). Porém, nesta dissertação, considerou-se que os alvos poderiam se movimentar com uma aceleração constante, permitindo uma análise nos efeitos que o parâmetro de aceleração exercem na performance do rastreador.

O algoritmo 1, na seção 2.6.1, foi o mesmo usado neste rastreador, mas os detalhes da implementação são discutidos agora. O

primeiro passo é assumir um comportamento linear para o rastreamento em uma aplicação de rastreamento. Considerando um objeto que se movimenta em uma imagem com aceleração constante e que o sensor é a saída de um detector de objetos. Pode-se representar o movimento de um objeto nos eixos x e y da imagem pelas equações em (3.3) e (3.4), onde $s_{0(x)}$ e $s_{0(y)}$ são as posições iniciais, $v_{0(x)}$ e $v_{0(y)}$ são as velocidades anteriores e a_x e a_y as acelerações em x e y na imagem.

$$x_t = s_{0(x)} + v_{0(x)} * dt + \frac{a_x * dt^2}{2} \quad ; \quad y_t = s_{0(y)} + v_{0(y)} * dt + \frac{a_y * dt^2}{2} \quad (3.3)$$

$$v_{t(x)} = v_{0(x)} + a_x * dt \quad ; \quad v_{t(y)} = v_{0(y)} + a_y * dt \quad (3.4)$$

Deve-se transformar essas equações em espaço de estados. Se assumirmos que X_t é um estado atual de um objeto que indica as posições e as velocidades nos dois eixos e assumirmos que um estado atual é resultado de uma mudança do estado anterior por uma ação de comando constante (u), as equações 3.3 e 3.4 podem ser representadas por:

$$X_t = A * X_{t-1} + B * u(t) + \omega \quad (3.5)$$

onde ω é um ruído do processo, A é uma matriz de transição de estados e B é uma matriz que relaciona a ação de comando com os estados iniciais (3.6).

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ; \quad B = \begin{bmatrix} 0.5 * dt^2 \\ 0.5 * dt^2 \\ dt \\ dt \end{bmatrix} \quad (3.6)$$

Agora que se está trabalhando no espaço de estados, podemos assumir que os dados de entrada do detector de objetos são resultantes dos estados referentes às posições x e y do objeto e alguma insegurança sobre essa medida referente ao processo de amostragem. Logo, pode-se modelar o sensor como na equação (3.7), onde C é uma matriz de correlação entre estados e leitura.

$$Z_t = C * X_t + \sigma \quad ; \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (3.7)$$

O filtro de Kalman trabalha com matrizes de covariância em seus cálculos. Nelas são representadas as incertezas sobre o processo (Q) e sobre os dados oriundos dos sensores (R). Como estamos modelando posição e velocidade, precisamos medir o efeito dos ruídos sobre esses modelos com base na ação de comando, que neste caso é a aceleração.

Se considerarmos que tanto no processo quanto na leitura, as variáveis que lidam com ruído são independentes em cada eixo, as matrizes com esses ruídos podem ser expressas como se segue em (3.8).

$$Q = \begin{bmatrix} dt^4/4 & 0 & dt^3/2 & 0 \\ 0 & dt^4/4 & 0 & dt^3/2 \\ dt^3/2 & 0 & dt^2 & 0 \\ 0 & dt^3/2 & 0 & dt^2 \end{bmatrix} * \omega ;$$

$$R = \begin{bmatrix} \sigma_x * \sigma_x & \sigma_x * \sigma_y \\ \sigma_y * \sigma_x & \sigma_y * \sigma_y \end{bmatrix} = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} \quad (3.8)$$

Os valores de parâmetros utilizados nos algoritmos propostos foram obtidos através de testes de performance e podem ser encontrados na seção 4.2.2.

3.2.3 Associação de dados

Depois de estimados os estados futuros dos objetos, duas heurísticas distintas podem ser adotadas para tratar a questão de fusão de *blobs*. O RAMUK realiza uma pré-associação entre as informações de posição dessas previsões com a posição dos alvos candidatos vindos da análise de *blobs*. Essa pré-associação usa o *hungarian algorithm* para atribuir uma previsão de trilha ativa a uma detecção mais próxima. Em seguida, se compara a área atual desse *blob* com a área anterior do objeto associado: caso a área seja maior que 60% da última área registrada, esse aumento é interpretado como uma fusão entre dois *blobs* e então as associações envolvidas são desfeitas e o *blob* é ignorado.

Já o RAMUK-DL utiliza as informações do FK, aumenta um pouco a área das projeções do *bounding box* de cada previsão por um parâmetro, chamado *offset*, e verifica se há sobreposições entre as projeções. Se este for o caso, os *blobs* que fazem intersecção com os *bounding boxes* são ignorados e não passam para a associação pelo *hungarian algorithm*.

Como podem haver situações nas quais objetos muito próximos podem receber identificações trocadas, o RAMUK-DL faz uso também da comparação entre descritores locais para ajudar a compensar uma associação mal feita pelo *hungarian algorithm*.

Tanto o RAMUK quanto o RAMUK-DL usam o FK no tratamento de oclusões. Quando um objeto não é detectado, ou seja, quando não existe uma detecção associada a uma trilha ativa, a posição do objeto é inferida apenas pela previsão. No caso do RAMUK-DL, a

previsão é combinada com as informações de *matching* do FREAK e a posição do objeto, $p(x, y)$, é inferida via média ponderada, conforme a equação (3.9)

$$p(x, y) = \frac{\alpha * p_{FREAK}(x, y) + (10 - \alpha) * p_{FK}(x, y)}{10} \quad (3.9)$$

onde, α é o número de pontos-chave combinados (10 é o valor máximo de α), $p_{FK}(x, y)$ é a posição prevista pelo FK, $p_{FREAK}(x, y)$ é a mediana das posições do centroide inferidas pela soma da posição de um ponto-chave e a distância registrada entre ele e o centroide do objeto ao qual ele foi relacionado no quadro anterior. O cálculo de $p_{Freak}(x, y)$ pode ser representado pela expressão (3.10):

$$p_{FREAK}(x, y) = \text{mediana} (\{(d_{x1}, d_{y1}) + p_1(x, y), \dots, (d_{xn}, d_{yn}) + p_n(x, y)\}) \quad (3.10)$$

onde (d_{xn}, d_{yn}) é a distância, em um quadro anterior, entre um ponto-chave e o centroide do objeto a que ele foi associado e $p_n(x, y)$ é a posição atual do *keypoint*.

Após o processo de associação, as trilhas associadas, não associadas e detecções não associadas são encaminhadas para o gerenciamento de trilhas.

3.3 GERENCIAMENTO DE TRILHAS

O gerenciamento de trilhas é responsável por inserir, penalizar e excluir as trilhas ativas. As detecções não associadas são interpretadas como novos objetos, logo são criadas novas trilhas para elas.

A atribuição de penalidades é feita sobre as trilhas não associadas, que podem receber penas de 1 ou 3 unidades. As penas unitárias são contabilizadas para as trilhas não associadas que tem seu *bounding box* sobreposto a alguma detecção. As trilhas não associadas cujos *bounding boxes* não estão sobrepostas a nenhuma detecção, recebem uma penalidade de 3 unidades. Já para as trilhas associadas, as penalidades são zeradas.

Caso os objetos estejam saindo do campo de visão da câmera, as trilhas associadas a esses objetos recebem o valor de pena máxima para que esta seja marcada para exclusão.

Depois de atribuir as penas quando necessário, os algoritmos marcam as trilhas que atingiram o limite de penalidades como trilhas não-ativas, sendo estas excluídas no final da iteração do algoritmo.

3.4 ARMAZENAMENTO E EXTRAÇÃO DE METADADOS

As informações das trilhas vindas do gerenciamento devem ser armazenadas pelo algoritmo para uso em próximas iterações e para serem exportadas como dados de saída do rastreador. As informações dos objetos são armazenadas em estados conforme explicito na seção 3.2.1.

Os metadados são exportados como uma matriz, na qual cada linha representa um objeto. As colunas registram: número do frame, identificação do objeto, posição do centroide em x, posição do centroide em y, altura e largura do *bouding box*. A figura 3.4 mostra um exemplo de metadados (variável de saída) do RAMUK-DL para dois quadros (75 e 76), nos quais foram registrados os estados de 5 objetos rastreados.

75	1	283.2962	205.0836	26	67
75	2	396.9618	225.3141	56	84
75	3	456.0035	233.2121	24	85
75	4	492.2442	226.9233	24	72
75	5	595.2915	234.7085	33	67
76	1	282.7474	204.9982	27	70
76	2	396.9817	225.3040	57	84
76	3	455.8212	232.7406	23	82
76	4	492.2836	226.6482	26	71
76	5	595.0209	234.4178	32	63

Figura 3.4 – Exemplo de dados de saída do RAMUK-DL

4 MÉTRICAS DE AVALIAÇÃO, DEFINIÇÕES DE PARÂMETROS E PLATAFORMA DE TESTES

Os algoritmos desenvolvidos neste trabalho utilizam parâmetros que, além de garantir o funcionamento do rastreador, realizam um ajuste fino do mesmo afim de se aperfeiçoar resultados. Alguns parâmetros de técnicas utilizadas neste trabalho mantiveram os valores definidos pelos seus autores – como os parâmetros da extração ViBe, enquanto outros parâmetros tiveram que ser estipulados de forma empírica, analisando os comportamentos e resultados das variáveis de saída.

A natureza aleatória de partes do algoritmo de rastreamento como a extração ViBe e a associação de dados por descritores podem gerar resultados pouco distintos para os mesmos valores de parâmetros. Logo, os vídeos utilizados foram testado por cinco vezes sob mesmas condições para se fazer uma média das métricas, tanto para expressar o resultado final de performance quanto para escolher valores de parâmetros.

Este capítulo discorre sobre como foram feitos os testes para a determinação dos valores de parâmetros dos algoritmos propostos. Também são abordadas as métricas utilizadas para avaliar os resultados dos testes, bem como as métricas usadas para avaliar as performances do RAMUK e RAMUK-DL depois de se definir os parâmetros.

4.1 MÉTRICAS PARA AVALIAÇÃO DE PERFORMANCE

Ao se aprofundar no problema de rastreamento de objetos nota-se que, a maioria das propostas publicadas antes do ano 2000 oferecem soluções para o rastreamento de um único objeto, sendo as métricas de performance mais utilizadas: precisão, *recall*, medida F-1, e número de sucessos. Entretanto, quando o foco é rastrear mais de um objeto em uma imagem, essas medidas são insuficientes para representar precisão, acurácia e a qualidade do rastreamento.

Segundo Bernardini e Stiefelhaven (2008), até o ano daquela publicação, não existia um consenso na comunidade de visão computacional sobre qual conjunto de medidas adotar para avaliar a performance de rastreadores de múltiplos objetos. Explorando esta oportunidade, Bernardini e Stiefelhaven (2006) propuseram a primeira versão do *CLEAR MOT Metrics*; um sistema de métricas que hoje é bem conhecido pelos pesquisadores para se avaliar rastreadores de múltiplos alvos.

Para calcular essas métricas é preciso criar um mapeamento entre objetos e hipóteses. Entende-se por objeto as anotações contidas no

gabarito (*ground truth*), enquanto as hipóteses são as saídas do rastreador. As métricas desse sistema são:

- *Taxa de falsos positivos (FP)* – faz referência ao número de hipóteses que não foram associadas a nenhum objeto. Pode estar associada a sensibilidade do detector de objetos que classifica erroneamente uma área como pertinente a um alvo. O cálculo desta métrica é indicado pela equação (4.1), onde fp_t é a quantidade de falsos positivos no instante t e g_t é a quantidade de objetos no mesmo instante;

$$\overline{FP} = \frac{\sum_t fp_t}{\sum_t g_t} \quad (4.1)$$

- *Taxa de perda ou taxa de falsos negativos (FN)* – essa taxa expressa os objetos que não foram rastreados pelo algoritmo. A equação (4.2) formaliza o cálculo desta métrica, onde fn_t é o número de objetos em um quadro t que não foram associados a nenhuma hipótese;

$$\overline{FN} = \frac{\sum_t fn_t}{\sum_t g_t} \quad (4.2)$$

- *Trocas de rótulos (EC)* – também conhecida como erro de correspondência (*mismatch error*), é uma métrica que representa a quantidade de vezes em que relações entre objetos e hipóteses sofrem alterações. Em outras palavras, se um objeto x , que se relaciona com uma hipótese y , passa a se relacionar com outra hipótese, a relação no mapa deve ser alterada com a nova relação e um erro de troca de rótulos deve ser computado. Ao final da avaliação de performance, esse acumulado é transformado em uma taxa (4.3) que é utilizada para o cálculo da MOTA;

$$\overline{EC} = \frac{\sum_t ec_t}{\sum_t g_t} \quad (4.3)$$

- *MOTA (Multiple Object Tracking Accuracy)* – esta métrica faz referência a acurácia do algoritmo e contempla as falhas detectadas durante o rastreamento. Pode ser decomposta em três

taxas de erros: a taxa de falsos positivos (4.1), taxa de falsos negativos (4.2) e taxa de erros de correspondência (4.3). O cálculo da MOTA é computado pela equação 4.4;

$$MOTA = 1 - (\overline{FP} + \overline{FN} + \overline{EC}) \quad (4.4)$$

- *MOTP (Multiple Object Tracking Precision)* – é uma métrica de precisão do rastreador e seu cálculo é expresso pela equação (4.5), onde i é o número de objetos existentes no quadro (t), c_i é o número de correspondências encontradas naquele quadro e $d_{i,t}$ é a diferença entre o objeto e sua hipótese correspondente. De acordo com Bernardin e Stiefelhagen (2008), para rastreadores de pessoas e veículos em 2D, essa diferença se refere à sobreposição entre a área do objeto e a área da hipótese, por isso essa métrica está fortemente associada à detecção de objetos.

$$MOTP = \frac{\sum_{i,t} d_{i,t}}{\sum_i c_i} \quad (4.5)$$

Posteriormente, surgiram outros parâmetros para avaliar a qualidade do rastreamento (Leal-Taixé *et al.*, 2015), sendo os principais: *Mostly Tracked* (MT), que representa a porcentagem das trilhas que são rastreadas em mais de 80% de sua vida útil; *Mostly Lost* (ML), que representa a porcentagem das trilhas que são rastreadas em até 20% de sua duração; *Partially Tracked* (PT), que representa a porcentagem das trilhas que são rastreadas entre 20% e 80% de sua vida útil.

Uma vez que o RAMUK e o RAMUK-DL foram projetados para acompanhar a trajetória de múltiplos alvos simultaneamente, adotou-se o *CLEAR MOT Metrics* (BERNARDIN; STIEFELHAGEN, 2008) para avaliar suas performances.

Mesmo não sendo um foco desta dissertação, foi mensurado o tempo de processamento de cada frame nos algoritmos RAMUK e RAMUK-DL, considerando as opções de exibição ou não do rastreamento. As medidas de performance de processamento foram expressas em quadros por segundo (*fps*) e foram obtidas contabilizando o número de quadros total dos vídeos, dividido pelo tempo total de execução de todos os quadros.

4.2 MÉTRICAS PARA AVALIAÇÃO DE PERFORMANCE

Os parâmetros de um sistema tem um importante papel no desempenho do mesmo. Isso implica que a escolha de seus valores deve ser levada em consideração durante o *design* da solução.

Mesmo que, em uma etapa inicial, os valores de parâmetros possam ser definidos por uma base teórica, o ajuste fino desses parâmetros pode ser determinante para se obter uma performance ainda melhor. Esses parâmetros podem ser estimados através de modelos de regressão, como o RANSAC, ou inferidos de forma empírica.

Quando se adota uma abordagem empírica, uma série de premissas e adaptações no código devem ser feitas para se conceber uma plataforma de testes.

Nesta seção são descritos os parâmetros usados pelo algoritmo proposto. Eles estão agrupados de acordo com a etapa em que são necessários. Também são fornecidos detalhes sobre os procedimentos de testes e métricas usadas para se estimar os melhores valores para estes parâmetros.

4.2.1 Parâmetros do ViBe

Os valores adotados para os parâmetros do ViBe foram os mesmos que aqueles usados por seus autores. Foram utilizadas 20 amostras para compor o modelo de plano de fundo. O raio da esfera de pertinência de um *pixel* ao modelo de plano de fundo foi de 20 unidades. O valor do *pixel* deve estar dentro do raio de pertinência de 2 modelos, no mínimo, para que ele seja considerado como plano de fundo. E por fim, a taxa de subamostras, na qual cada *pixel* classificado como plano de fundo tem 1 chance em 16 de seu valor ser atribuído a um mesmo *pixel* de um dos modelos de plano de fundo, além de 1 chance em 16 de atualizar o valor de um *pixel* vizinho de um dos modelos.

4.2.2 Parâmetros do Filtro de Kalman

Para estipular os melhores parâmetros de um Filtro de Kalman é necessário investigar os efeitos que diferentes parâmetros têm sobre o rastreamento. Entretanto, a inconsistência na detecção de objetos usando o ViBe torna a investigação dos efeitos do Filtro de Kalman nebulosa, pois a detecção de objetos exerce um impacto relevante sobre o rastreamento. Dessa forma, foi utilizada a mesma saída do ViBe durante os testes de parâmetros do FK para anular as perturbações causadas pelo segmentador de imagens.

Além da extração ViBe, outra etapa do algoritmo que pode interferir nos resultados dos testes de parâmetros do FK é a associação por descritores. Como não se investigou as implementações dos algoritmos utilizados para detecção de pontos de interesse e geração de descritores, foi sensato analisar os resultados do rastreamento utilizando apenas o FK, ou seja apenas o RAMUK foi usado para determinar o melhor valor para os parâmetros do FK. Foram testados: a entrada de controle (u), ruídos do processo (ω) e ruídos do sensor (σ).

A entrada de controle (u) e se refere à magnitude com a qual um objeto pode mudar sua trajetória de um quadro para outro. Segundo Cannons (2008), tanto a entrada de controle como a matriz de que relaciona essa entrada com os estados dos objetos são geralmente omitidas ao se usar o FK no rastreamento visual, considerando a movimentação dos objetos como sendo movimentos uniformes. Todavia, é interessante observar a saída do rastreador considerando que os objetos podem mudar sua aceleração e, portanto, tanto a entrada de controle quanto a matriz que a relaciona com os estados do objeto foram implementadas e o parâmetro u testado.

Os ruídos do sensor (σ) se refere às perturbações no processo de detecção de objetos. Como os dados de desvio padrão do sensor não são conhecidos, os valores para esse ruído tiveram que ser inferidos empiricamente. Os ruídos do processo (ω) são perturbações que afetam a transição de estados dos objetos e podem ser associados às inconsistências de modelagem. Após análise dos resultados (seção 5.1), os melhores parâmetros para o FK foram $u=0$, $\omega=1$ e $\sigma=5$.

4.2.3 Parâmetros da análise de *blobs*

Uma parte da detecção dos objetos é a análise de componentes conectados. Nesta etapa é possível definir qual é a altura mínima dos objetos de interesse. Essa parâmetro muda de sequência para sequência, uma vez que a dimensão dos objetos na imagem dependem de seu tamanho real e da distância entre ele e a câmera.

O algoritmo permite ao usuário determinar uma altura mínima de até um *pixel*, entretanto valores baixos para este parâmetro (abaixo de 10 *pixels*) permitem o rastreamento de falsos positivos ou até mesmo partes de objetos de interesse, atribuindo a um objeto mais de uma detecção. Por outro lado, uma altura mínima alta pode descartar muitas detecções, aumentando o número de falsos negativos.

A solução para este problema foi variar os valores de altura mínima (começando de 10 até 50 *pixels* em intervalos de 10 *pixels*) e

comparar os resultados médios de MOTA e MOTP para as bases de dados utilizada neste trabalho. Foram selecionados 4 vídeos do CAVIAR para representar todo este *dataset*, e a sequência S2L1 do PETS. Cada vídeo foi testado 5 vezes, variando os valores, totalizando 225 testes. Concluiu-se que a melhor altura mínima foi de 30 *pixels*. Os resultados desses teste são discutidos na seção 5.1.

Um parâmetro que poderia ter sido implementado no detector de objetos é a altura máxima, contudo esse parâmetro poderia limitar o uso do sistema quanto a variedade de objetos detectados e/ou ao tipo de aplicação que ele poderia ser utilizado e por isso ele não foi explorado.

4.2.4 Parâmetros para tratamento de junções de *blobs*

O *offset* foi um parâmetro adotado para lidar com a questão da junção de componentes conectados. Imagine que um objeto no instante t , ou sua projeção para o instante $t+1$ estejam sobrepostos ou muito próximas a outros; isso é um indicativo de que estes objetos segmentados (*blobs*) estão fundidos ou podem se fundir no próximo quadro; neste caso, deve-se ignorar a detecção e realizar o rastreamento através do FK e descritores. O *offset* aumenta a área dos *bounding boxes* e permite verificar uma sobreposição entre objetos antes que os *blobs* se conectem uns aos outros.

Se por um lado, um *offset* pequeno não impede a detecção de um *blob* fundido, por outro lado um valor alto deste parâmetro pode ignorar muitas detecções. Os testes para estipular o *offset* ideal seguiram o mesmo conceito dos testes que definiram a altura mínima e seu valor foi estipulado em 5 *pixels*.

4.2.5 Parâmetros na associação de dados

O RAMUK-DL utiliza descritores para ajudar na associação entre detecções e trilhas, além de usá-los para estimar posições dos objetos durante oclusões. Basicamente, se compara descritores de um objeto em quadros anteriores com os descritores encontrados no quadro atual. Esses descritores devem ser armazenados em um “saco de descritores” e devem permanecer disponíveis por um curto intervalo de tempo para se evitar um alto custo computacional comparando muitos descritores.

Dessa forma, o parâmetro duração dos descritores determina por quantos quadros os descritores extraídos em um quadro anterior devem existir. Durante a implementação do rastreador, foi observado que descritores com duração maiores que 5 quadros geram poucas

combinações com os descritores do quadro atual e podem apontar combinações falsas. Portanto esse parâmetro foi definido a partir de observações qualitativas dos resultados.

4.2.6 Parâmetros para o gerenciamento de trilhas

O parâmetro de pena máxima pode ser visto como um relaxamento do rastreador. Ele permite que trilhas sem objetos associados possam continuar existindo por um tempo até que a detecção do objeto perdido seja detectada novamente e a relação entre trilha e detecção reestabelecida.

A cada quadro, uma trilha sem associação recebe a penalidade de uma unidade caso a projeção de seu *bounding box* esteja sobre algum *blob*. Caso contrário, a penalidade é de três unidades. A trilha deixa de existir se o número de penalidades superar o limite estabelecido.

Com uma pena máxima alta, trilhas não associadas podem permanecer na imagem por muito tempo, o que acarreta no aumento do número de falsos positivos e reduz a precisão do rastreador. Já uma pena máxima pequena elimina as trilhas antes que as mesmas possam se recuperar de uma perda; neste caso, quando um objeto que tinha sido perdido é detectado novamente, entende-se que ele é um objeto novo e uma nova trilha é atribuída a ele, acrescentando o número de trocas de rótulos.

Foram testados 4 vídeos do CAVIAR, representando todo este *dataset*, e a sequência S2L1 do PETS. Cada vídeo foi testado 5 vezes, variando os valores de pena máxima entre 3, 8 e 15. A pena máxima de 8 infrações obteve os melhores resultados considerando a média entre os *datasets*.

4.3 BASE DE DADOS E ANOTAÇÕES DE VÍDEOS (*DATASET*)

Via de regra, os pesquisadores que desenvolvem algoritmos de rastreamento testam a performance de seus algoritmos e comparam seus resultados com resultados obtidos por outros algoritmos propostos na literatura.

Os resultados de performance sobre um vídeo são gerados a partir da comparação entre a saída de um rastreador e as anotações desse vídeo. Essas anotações, também conhecidas como *ground truth*, podem ser entendidas como um gabarito onde, quadro-a-quadro, estão registradas informações sobre os objetos como rótulos (números), posições de centro e dimensões de *bounding boxes*. Criar um gabarito desses é um trabalho minucioso e demorado, pois requer supervisão e ajustes manuais praticamente em todos os quadros de vídeo.

Com o intuito de facilitar a vida dos pesquisadores que estudam o rastreamento de objetos, algumas instituições ou autores disponibilizam seus conjuntos de dados (*datasets*) que contém sequências de vídeos em diversos contextos, bem como os devidos gabaritos para que os pesquisadores possam testar seus algoritmos.

Uma comparação de resultados relevante considera a performance dos algoritmos sob um mesmo *datasets*, respeitando as mesmas condições e usando as mesmas métricas para avaliação. Nesse sentido surgem as plataformas de avaliação (*benchmarks*) ou desafios (*challenges*) – como o PETS (*Performance Evaluation of Tracking and Surveillance*), o MOT Challenge (*MultiObject Tracking Benchmark*) e o VOT Challenge (*Video Object Tracking*) – que são trabalhos cujo objetivo é fornecer um panorama sobre os desempenhos de rastreadores que foram testados em seu conjunto de dados.

Submeter o algoritmo para desafios ou plataforma de testes não é a única forma de avaliar o desempenho dos algoritmos. Muitos autores escolhem uma ou mais bases de dados e procuram no estado da arte outros trabalhos que usaram o mesmo *dataset* para comparar os resultados.

O levantamento feito por Dubuisson e Gonzales (2016) sobre *datasets* para rastreadores de objetos é um bom ponto de partida para buscar um conjunto de vídeos que mais se adequam a determinada aplicação. Após analisar os *datasets* abordados pelos autores com relação à disponibilidade dos vídeos e anotações, à quantidade de objetos anotados e à similaridade com imagens de um CFTV, foram pré-selecionados os conjuntos de vídeos do CAVIAR, INRIA e PETS (2009).

O CAVIAR (*Context Aware Vision using Image-based Active Recognition*) tem sido utilizado em trabalhos de análise comportamental e reconhecimento de ações humanas. As imagens disponíveis nesse *dataset* são feitas dentro de estabelecimentos (*indoor*) e disponibilizam mais de um ponto de vista (*multi-view*). Contudo, o CAVIAR não disponibiliza imagens obtidas em ambientes abertos (*outdoor*), como ruas e praças, que geralmente oferecem um desafio maior para os rastreadores de objetos e permite avaliar o desempenho deles em situações mais gerais.

O INRIA (*Institut National de Recherche en Informatique et en Automatique*) é uma coletânea de vídeos e imagens usadas nas pesquisas de Dalal e Triggs (2005) e Dalal (2006) para a detecção de pessoas em pé. Possui sequências de longa duração e as anotações fornecidas não contemplam objetos menores que 100 *pixels*. Infelizmente não foram encontrados muitos trabalhos publicados sobre este *dataset*.

O PETS, além de ser um *benchmark*, disponibiliza também seus *datasets* compostos por vídeos *multi-view* de ambientes externos, para avaliação de desempenho de rastreadores. Isso torna o uso do PETS atraente aos pesquisadores, pois é possível para testar seus algoritmos e comparar seus resultados com os resultados de uma gama de outros rastreadores publicados na literatura.

Além da similaridade dos vídeos com a aplicação em foco, a escolha do conjunto de vídeos também deve considerar a quantidade de artigos que utilizam esses *datasets* para avaliar algoritmos, pois não é usual encontrar trabalhos que disponibilizem pseudocódigos ou informações suficientemente claras para recriar os algoritmos e testá-los em vídeos diferentes para se comparar os resultados.

Sendo assim, buscou-se na literatura, algoritmos ou levantamentos comparativos (*benchmarks*) sobre rastreamentos de múltiplos objetos que utilizaram os *datasets* pré-selecionados. Os conjuntos de vídeos com mais referências encontrados nas pesquisas foram as sequências do PETS-2009 e as sequências do projeto CAVIAR. Os vídeos escolhidos neste trabalho foram as 20 sequências da vista de um corredor de um shopping center em Lisboa (CAVIAR) e o vídeo S2L1(PETS-2009).

5 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Nesta seção são apresentados e discutidos os resultados dos testes realizados para definir os melhores parâmetros para o RAMUK e RAMUK-DL. Depois são apresentados e analisados os resultados de desempenho dos mesmos. No final dessa seção é feita uma comparação de desempenho considerando resultados de outros algoritmos publicados em cima do PETS-2009 S2L1 e o CAVIAR 2006.

5.1 RESULTADOS DOS TESTES DE PARÂMETROS

Visto que a MOTA é uma medida que representa erros de correspondência, falsos positivos e falsos negativos, as medidas adotadas como critérios para a escolha de parâmetros foram MOTA e MOTP.

Os testes para definir os parâmetros do FK foram feitos em uma adaptação do RAMUK para reduzir a influência da segmentação ViBe e dos descritores locais na associação de dados. Os resultados completos podem ser vistos no Apêndice I.

Para analisar o parâmetro u foram calculados os valores médios de MOTA e MOTP, considerando todas as variações dos parâmetros ω e σ em ambos *datasets*. Os melhores resultados estão expressos em negrito.

Tabela 1 – Análise do parâmetro u

DATASETS	MÉDIAS (%)	PARÂMETRO (u)			
		0	0,01	0,1	1
PETS 2009	MOTA	84,65	84,64	84,59	67,02
	MOTP	62,89	62,86	62,14	52,19
CAVIAR2006	MOTA	43,88	43,84	43,23	35,95
	MOTP	63,07	63,13	62,58	52,65
MÉDIAS DE MOTA E MOTP DE AMBOS DATASET		63,62	63,62	63,14	51,95

Observando a tabela 1, nota-se que, assumir um movimento praticamente uniforme para os objetos ($u \rightarrow 0$) levou o RAMUK aos melhores resultados. Logo, adotou-se uma ação de controle nula na modelagem do sistema. Considerando os resultados gerados com o parâmetro $u=0$, a escolha de ω e σ foi feita com base em uma média

ponderada entre os *datasets*, pois o volume de vídeos do CAVIAR é muito maior que o número de vídeos do PETS. Para compensar essa diferença, foi atribuído um peso maior (0,7) para os vídeos da sequência do PETS.

Tabela 2 – Comparação dos parâmetros ω e σ

MEDIDAS	PARÂMETRO ω	PARÂMETRO σ					
		1	5	10	15	20	30
MOTA PONDERADA	0,5	73,80	73,86	72,66	71,48	69,73	64,45
	1	73,01	74,41	73,86	72,89	72,66	71,48
	5	69,70	73,04	73,66	74,19	74,34	74,23
	10	68,39	73,01	73,04	73,34	73,64	74,19
	15	67,32	72,46	73,46	73,04	73,53	73,67
MOTP PONDERADA	0,5	64,81	64,02	60,17	57,74	55,94	52,93
	1	64,75	65,29	63,02	61,49	60,17	57,74
	5	63,80	65,72	64,84	64,75	64,51	63,94
	10	63,87	65,98	64,72	64,85	64,85	64,74
	15	64,02	65,12	65,07	64,73	64,82	64,84
MÉDIAS ENTRE MOTA E MOTP	0,5	69,31	68,94	66,41	64,61	62,83	58,69
	1	68,88	69,85	68,44	67,19	66,41	64,61
	5	66,75	69,38	69,25	69,47	69,43	69,08
	10	66,13	69,49	68,88	69,10	69,24	69,47
	15	65,67	68,79	69,27	68,88	69,18	69,26

Pela tabela 2, podemos inferir que os melhores valores para ω e σ que são respectivamente 1 e 5. Esses valores fazem sentido, pois ao se assumir valores altos para ω faz com que o rastreador confie demais nas detecções e quando essa falha, o filtro de Kalman aponta valores muito distintos das posições reais. Um valor baixo desse parâmetro acaba anulando a contribuição dos sensores na estimativa de posição. Similarmente, σ baixo leva a uma baixa confiança nos dados dos sensores, mas um valor alto para este parâmetro pode levar a perda a trilha em caso de uma detecção errônea.

Com os parâmetros do FK definidos, inicia-se a busca pela altura mínima dos objetos de interesse. Esse parâmetro pertence à etapa análise de *blobs* e é útil para eliminar detecções indesejadas. Devido à natureza aleatória do ViBe, cada vídeo selecionado foi testado 5 vezes e a média dos resultados para MOTA e MOTP foi utilizada para análise da melhor altura.

Tabela 3 – Resultados para o teste de altura mínima

DATASETS	MÉDIAS	PARÂMETRO ALTURA MÍNIMA (pixels)				
		10	20	30	40	50
PETS2009	MOTA	79,99	84,05	86,65	89,69	91,10
S2L1	MOTP	50,24	53,92	65,88	49,16	48,27
CAVIAR 2006	MOTA	50,24	53,92	65,88	49,16	48,27
EnterExitCrossingPaths1cor	MOTP	44,15	48,87	48,48	41,19	46,06
CAVIAR 2006	MOTA	20,10	24,39	28,82	29,65	33,33
EnterExitCrossingPaths2cor	MOTP	60,01	70,37	52,33	74,49	70,58
CAVIAR 2006	MOTA	34,31	36,30	25,77	15,31	27,88
OneLeaveShopReenter2cor	MOTP	44,57	45,90	47,73	54,82	58,58
CAVIAR 2006	MOTA	59,14	56,93	65,52	66,41	65,17
OneStopNoEnter1cor	MOTP	59,83	63,68	64,69	60,99	72,38
MÉDIAS PARA OS VÍDEOS DO CAVIAR 2006	MOTA	40,95	42,89	46,49	40,13	43,66
	MOTP	52,14	57,21	53,31	57,87	61,90
MÉDIAS DAS MÉDIAS DE MOTA E MOTP		55,83	59,52	63,08	59,21	61,23

A tabela 3 é um resumo dos resultados expostos no Apêndice II. De acordo com essa tabela, uma altura mínima de 30 *pixels* pode gerar os melhores resultados em média para MOTP e MOTA considerando os dois *datasets*.

O RAMUK-DL utiliza o *offset* em uma heurística que trata a questão da fusão de *blobs*. A tabela 4 ilustra os valores médios de MOTA e MOTP de cada vídeo selecionado para o teste; segundo essa tabela, um *offset* de 5 *pixels* mostrou-se mais adequados para considerando os 2 *datasets*. Os resultados completos estão no Apêndice III.

Os efeitos de um *offset* grande é que muitas detecções podem ser ignoradas, mesmo que as projeções de suas trilhas não estejam sobrepostas umas às outras. Logo, a trilha que estava associada a um objeto passa a ser inferida pelo FK, ao invés de usar dados bons de um detector de objetos. Em contrapartida, *offset* pequeno não impede que os *blobs* se juntem, fazendo com que apareça uma detecção que englobe mais de um objeto, o que afeta a acurácia e precisão do rastreador. Logo, o valor de 5 *pixels* está dentro do esperado para o *offset*.

Tabela 4 – Resultados para o teste de *offset*

SEQUÊNCIAS	MÉDIAS	PARÂMETRO OFFSET (pixels)				
		0	3	5	7	10
PETS-2009	MOTA	85,64	86,48	86,99	88,00	84,68
S2L1	MOTP	60,85	61,81	62,05	61,75	59,72
CAVIAR 2006	MOTA	46,97	45,27	44,57	38,14	42,59
EnterExitCrossingPaths1cor	MOTP	40,27	38,25	45,17	38,27	42,25
CAVIAR 2006	MOTA	33,67	33,72	35,58	35,87	37,93
EnterExitCrossingPaths2cor	MOTP	69,72	67,67	67,02	67,01	67,45
CAVIAR 2006	MOTA	17,23	27,79	30,15	38,35	29,37
OneLeaveShopReenter2cor	MOTP	54,42	54,55	54,19	47,84	48,23
CAVIAR 2006	MOTA	54,89	55,52	61,30	62,37	61,39
OneStopNoEnter1cor	MOTP	76,47	74,46	76,85	75,51	67,45
MÉDIAS PARA OS VÍDEOS DO CAVIAR 2006	MOTA	38,19	40,58	42,90	43,68	42,82
	MOTP	60,22	58,73	60,81	57,16	56,34
MÉDIAS DAS MÉDIAS DE MOTA E MOTP		61,22	61,90	63,19	62,65	60,89

A pena máxima determina quando se deve excluir uma trilha ruim. O algoritmo foi testado considerando 3 valores para esse parâmetro (3, 8 e 15). A análise dos resultados seguiu a mesma filosofia adotada para altura mínima e *offset*. Todos os dados deste teste estão no Apêndice IV, mas a tabela 5 resume esses dados. Com base nessa tabela, definiu-se que as trilhas que atingem o limite de 8 infrações devem ser eliminadas.

Tabela 5 – Resultados para o teste de pena máxima

SEQUÊNCIAS	MÉDIAS	PARÂMETRO PENA MÁXIMA		
		3	8	15
PETS-2009	MOTA	86,70	88,57	85,04
S2L1	MOTP	66,33	66,76	64,47
CAVIAR 2006	MOTA	47,07	44,39	43,26
EnterExitCrossingPaths1cor	MOTP	39,67	42,69	40,59
CAVIAR 2006	MOTA	35,97	31,25	28,42
EnterExitCrossingPaths2cor	MOTP	66,74	65,94	61,91
CAVIAR 2006	MOTA	28,21	29,96	25,01
OneLeaveShopReenter2cor	MOTP	52,71	56,43	47,25
CAVIAR 2006	MOTA	54,89	55,86	61,30
OneStopNoEnter1cor	MOTP	74,70	64,89	70,41
MÉDIAS PARA OS VÍDEOS DO CAVIAR 2006	MOTA	41,54	40,36	39,50
	MOTP	58,45	57,49	55,04
MÉDIAS DAS MÉDIAS DE MOTA E MOTP		63,25	63,30	61,01

5.2 RESULTADOS DE PERFORMANCE

Após definir os parâmetros dos algoritmos RAMUK e RAMUK-DL, ambos passaram por uma bateria de testes. Cada vídeo escolhido foi testado 10 vezes para verificar a acurácia (MOTA), precisão (MOTP), taxa de Falsos Positivos (*FP*), taxa de Falsos Negativos (*FN*), Erros de Correspondência (*EC*), *Mostly Tracked* (*MT*), *Partially Tracked* (*PT*) e *Mostly Lost* (*ML*). A tabela 6 contém as médias dos resultados dos testes de performance por *dataset*. Os resultados para todas as seqüências podem ser consultados no Apêndice V.

Tabela 6 – Médias dos resultados de performance para RAMUK e RAMUK-DL

ALGORITMO	DATASET	MOTP (%)	MOTA (%)	FP (%)	FN (%)	EC	MT (%)	PT (%)	ML (%)
RAMUK	PETS	68,08	87,29	2,65	8,71	53	89,47	10,53	0,00
RAMUK-DL	2009	69,34	90,48	1,83	7,10	29	84,21	15,79	0,00
RAMUK	CAVIAR	62,62	38,56	6,74	51,58	20	30,28	41,39	28,33
RAMUK-DL	2006	64,55	38,65	6,72	50,53	25	30,58	40,38	29,04

Ao se comparar os resultados dos algoritmos propostos, nota-se que o RAMUK-DL apresentou uma leve vantagem em média sobre o RAMUK. Isso mostra que o uso dos descritores e heurísticas para tratamento de fusão de *blobs* aumentam a precisão e reduziu bastante os erros de correspondência melhorando a acurácia do rastreamento em ambos os *datasets*.

Outro fato interessante foi a grande diferença de desempenho de um *dataset* para outro. Tanto o RAMUK quanto o RAMUK-DL não apresentam boa performance nos vídeos do CAVIAR. Pode-se dizer que essa diferença é decorrente de dois motivos: longas sobreposições de objetos e seqüências de curta duração.

No CAVIAR, as seqüências foram feitas por uma câmera cujo campo de visão está quase em paralelo com o comprimento de um corredor. Isso favorece o aparecimento e/ou permanência de objetos sobrepostos por um longo tempo e as heurísticas adotadas para lidar com a fusão de *blobs* falham, levando a uma detecção de objetos ruim. Além dos falsos negativos e perda de precisão gerados pela detecção de *blobs* fundidos, o *ground truth* do CAVIAR anota posição de algumas pessoas em tamanhos pequenos no final. Como a altura mínima foi estipulada em 30 *pixels*, qualquer objeto com altura menor que essa é ignorado, gerando mais falsos negativos e reduzindo a MOTA.

O tempo curto das sequências interfere na construção de um modelo de plano de fundo mais confiável, pois quanto maior o vídeo, melhor o ViBe consegue diferenciar objetos e plano de fundo, afetando a detecção de objetos. Ademais, uma sequência curta favorece o aparecimento de “fantasmas”, como ilustrado na figura 5.1. Um modelo de plano de fundo mal feito aumenta o número de falsos positivos, falsos negativo se reduz a precisão do rastreador.

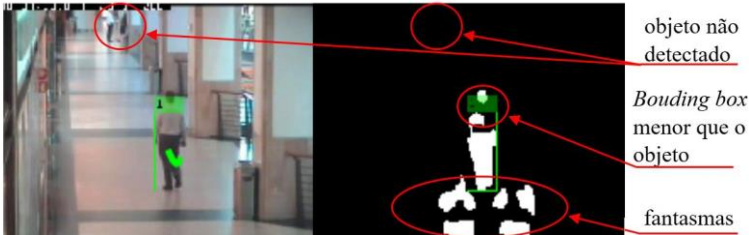


Figura 5.1 – Segmentação ViBe com fantasmas, objetos não detectados e objetos com tamanho errado.

Outro ponto que contribuiu para que os melhores resultados fossem atingidos na sequência do PETS foi que nela não há reflexos dos objetos em vidros e pisos reflexivos, como ocorrem nas sequências do CAVIAR. Isso ficou notório na análise qualitativa dos resultados, que por muitas vezes, indicou rastreamento de imagens refletidas como falsos positivos (afetando a MOTA) ou aumentou a área dos objetos detectados, diminuindo o MOTP. A figura 5.2 ilustra esses problemas: o objeto 1 teve seu *bounding box* aumentado pelo seu reflexo no piso; o objeto 2 se trata de um fantasma; o objeto 3 é referente a um reflexo do alvo 1 no vidro da loja; por fim existem três alvos não detectados, pois seus *pixels* foram incorporados no modelo de plano de fundo do ViBe.



Figura 5.2 – Exemplos de problemas no rastreamento em amarelo

A análise qualitativa permite avaliar o funcionamento do rastreador quanto a alguns de seus parâmetros e heurísticas, como por exemplo: a figura 5.3 mostra uma detecção sendo ignorada, pois ela não apresenta a altura mínima estipulada; a figura 5.4 mostra uma falha na heurística que trata a fusão de *blobs*; o tratamento de oclusões e tratamento de fusão de *blobs* pode ser observado pela figura 5.5; na figura 5.6 é ilustrado o *matching* de pontos-chaves de um objeto entre quadros através de descritores FREAK.

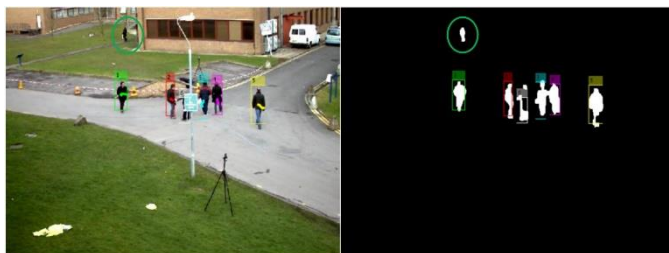


Figura 5.3 – Detecção ignorada pelo parâmetro de altura mínima



Figura 5.4 – Fusão de *blobs* não evitada

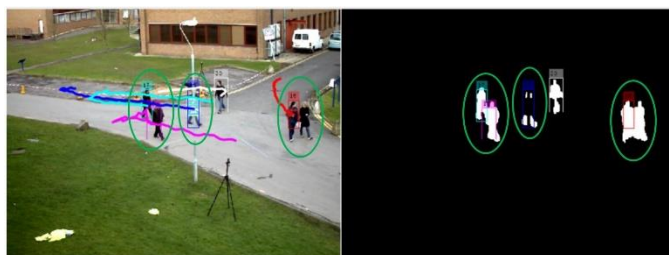


Figura 5.5 – Tratamento de fusão de *blobs* e de oclusão de objetos

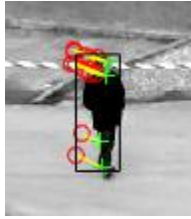


Figura 5.6 – Associação de pontos-chave entre quadros através do FREAK

Em um vídeo feito para testar a robustez do rastreador quanto à mudança de iluminação repentina, foi observado que o ViBe conseguiu se recuperar e voltou a detectar o objeto, observe o resultado qualitativo na figura 5.6.

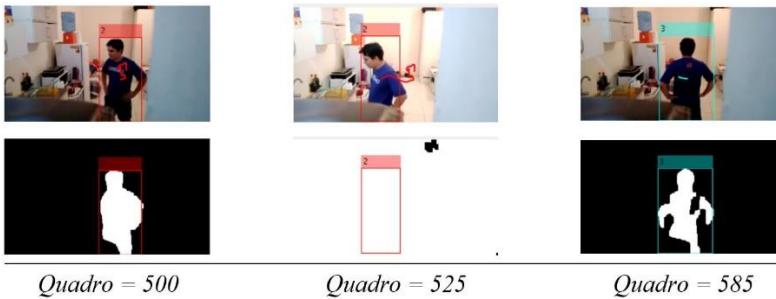


Figura 5.7 – Variação de iluminação repentina e recuperação da detecção

As performances de processamento dos algoritmos propostos foram mensuradas na forma de taxa de quadros processados por segundo (*fps*). Para realizar os testes dos algoritmos propostos foi utilizado um notebook LENOVO Idealpad Y510P, dotado com um processador Intel® core i7® 4700MQ 2.4 GHz, memória de 8 GB RAM DDR3, 2 placas de vídeo NVIDIA® GeForce® GT750M de 2 GB, SSD Kingston 250 GB e Windows® 10 Pro. Tanto os algoritmos quanto seus testes foram executados em MATLAB® R2015a. A tabela 7 contém os dados de performance de processamento. Quanto maior o valor, mais rápido é o processamento.

Como o RAMUK é um algoritmo mais simples que o RAMUK-DL, era esperado e foi constatado que o primeiro apresentasse a maior taxa de processamento em todos os vídeos testados, exibindo ou não o rastreamento. A diferença de processamento é bem significativa e deve ser levada em consideração na escolha de um algoritmo para uma aplicação de tempo real.

Tabela 7 – Taxa de processamento dos algoritmos RAMUK e RAMUK-DL sem e com exibição do rastreamento

VÍDEO	RESOLUÇÃO (Pixels)	TAXA DE EXECUÇÃO SEM EXIBIÇÃO (fps)		TAXA DE EXECUÇÃO COM EXIBIÇÃO (fps)	
		RAMUK	RAMUK-DL	RAMUK	RAMUK-DL
PETS09-S2L1	768 x 576	4,70	3,28	3,54	3,21
EnterExitCrossingPaths1cor	384 x 288	8,77	7,20	8,90	5,95
EnterExitCrossingPaths2cor	384 x 288	10,82	6,74	7,34	5,69
OneLeaveShop1cor	384 x 288	15,33	8,76	10,42	6,15
OneLeaveShopReenter2cor	384 x 288	10,96	6,12	6,99	5,36
OneShopOneWait1cor	384 x 288	9,38	5,67	6,73	4,96
OneShopOneWait2cor	384 x 288	9,26	5,27	6,55	4,81
OneStopEnter1cor	384 x 288	9,25	6,47	6,54	5,40
OneStopEnter2cor	384 x 288	9,03	6,41	6,48	5,47
OneStopMoveEnter1cor	384 x 288	9,61	4,83	5,53	4,44
OneStopMoveEnter2cor	384 x 288	9,19	6,77	6,68	5,67
OneStopMoveNoEnter1cor	384 x 288	9,58	6,95	7,07	5,78
OneStopNoEnter1cor	384 x 288	10,21	6,20	6,97	5,41
OneStopNoEnter2cor	384 x 288	9,51	5,93	6,51	5,19
ShopAssistant1cor	384 x 288	9,47	6,33	6,45	5,32
ShopAssistant2cor	384 x 288	8,98	5,68	6,40	5,00
ThreePastShop1cor	384 x 288	9,67	5,47	6,62	4,21
TwoEnterShop1cor	384 x 288	9,51	5,78	6,61	4,54
TwoEnterShop3cor	384 x 288	9,52	6,27	6,54	4,04
TwoLeaveShop2cor	384 x 288	11,67	7,67	7,24	4,60
WalkByShop1cor	384 x 288	9,22	6,00	6,53	4,49

5.3 COMPARATIVO COM OUTROS ALGORITMOS NA LITERATURA

Uma comparação de resultados relevante entre algoritmos deve ser feita sobre os mesmos vídeos. A tabela 8 apresenta uma comparação de resultados considerando a sequência S2L1 do PETS 2009 e a tabela 9 compara os dados sobre o CAVIAR 2006. Os 3 melhores resultados estão expressos em negrito.

Tabela 8 – Comparativo entre resultados sobre o PETS 2009 - S2L1

<i>ALGORITMO</i>	<i>MOTP</i> (%)	<i>MOTA</i> (%)	<i>FP</i> (%)	<i>FN</i> (%)	<i>EC</i>	<i>MT</i> (%)	<i>PT</i> (%)	<i>ML</i> (%)
Choi <i>et al.</i> , 2015	67,47	84,21	-	-	-	-	-	-
Breitenstein <i>et al.</i> , 2011	56,30	79,70	-	-	-	-	-	-
Kim <i>et al.</i> , 2015 - MHT	78,80	92,30	-	-	17	78,26	21,74	0
Kim <i>et al.</i> , 2015 - MHT-DAM	79,10	92,60	-	-	13	73,91	26,09	0
Cox e Hingorani, 1996 - MHT	77,50	84,10	-	-	45	78,26	21,74	0
Berclaz <i>et al.</i> , 2011	58,00	80,00	-	-	28	-	-	-
Shitrit <i>et al.</i> , 2011	58,38	81,46	-	-	19	-	-	-
Henriques <i>et al.</i> , 2011	68,74	88,77	-	-	10	-	-	-
RoshanZamir <i>et al.</i> 2012	69,02	90,30	-	-	8	-	-	-
Andriyenko e Schindler, 2011	73,93	81,84	-	-	15	-	-	-
Milan e Schindler, 2013	74,30	90,30	-	-	22	78,26	21,74	0
Milan <i>et al.</i> , 2014 - OM+APP	80,20	90,60	1,27	6,49	11	91,30	4,35	4,35
Milan <i>et al.</i> , 2014 - no OM	79,30	91,60	1,14	5,63	16	91,30	8,70	0,00
Milan <i>et al.</i> , 2014 - OM	76,90	88,60	5,57	3,68	19	91,30	8,70	0,00
Milan <i>et al.</i> , 2014 - EKF	76,50	68,00	1,40	25,23	25	39,13	56,52	4,35
Bae e Yoon, 2014 - p1	69,39	78,19	20,3	1,32	16	100	0	0
Bae e Yoon, 2014 - p2	69,01	77,38	20,41	2,18	10	100	0	0
Bae e Yoon, 2014 - p3	66,39	78,49	19,99	1,37	6	100	0	0
Bae e Yoon, 2014 - p4	69,59	83,04	19,41	1,19	4	100	0	0
RAMUK	68,08	87,29	2,65	8,71	53	89,47	10,53	0,00
RAMUK-DL	69,34	90,48	1,83	7,10	29	84,21	15,79	0,00

Uma comparação de resultados relevante entre algoritmos deve ser feita sobre os mesmos vídeos. A tabela 8 apresenta uma comparação de resultados considerando a sequência S2L1 do PETS 2009 e a tabela 9 compara os dados sobre o CAVIAR 2006. Os 3 melhores resultados estão expressos em negrito.

Tabela 9 – Comparativo entre resultados sobre o CAVIAR 2006

<i>ALGORITMO</i>	<i>MOTP</i> (%)	<i>MOTA</i> (%)	<i>FP</i> (%)	<i>FN</i> (%)	<i>EC</i>	<i>MT</i> (%)	<i>PT</i> (%)	<i>ML</i> (%)
Schumann <i>et al.</i> , 2013	-	60,26	19,80	34,22	-	-	-	-
Schumann <i>et al.</i> , 2013 - MGT	-	61,32	19,20	32,58	-	-	-	-
Huang <i>et al.</i> , 2008	-	80,00	2,50	20,00	-	-	-	-
Guan <i>et al.</i> , 2013 - B	35,53	41,48	-	-	-	-	-	-
Guan <i>et al.</i> , 2013 - C'	44,82	70,93	-	-	-	-	-	-
Guan <i>et al.</i> , 2013 - C'PR	53,66	87,86	-	-	-	-	-	-
Bae e Yoon, 2014 - p1	85,26	81,37	1,23	17,45	25	85,34	16,61	1,05
Bae e Yoon, 2014 - p2	84,36	80,15	1,59	18,26	21	77,49	14,66	7,85
Bae e Yoon, 2014 - p3	86,75	83,43	1,20	14,92	24	87,96	12,04	0
Bae e Yoon, 2014 - p4	87,15	86,52	1,00	11,38	9	89,53	10,47	0
RAMUK	62,62	38,56	6,74	51,58	20	30,28	41,39	28,33
RAMUK-DL	64,55	38,65	6,72	50,53	25	30,58	40,38	29,04

Infelizmente, não foram encontradas muitas contribuições que avaliam os algoritmos considerando o *CLEAR MOT Metrics* completo, portanto muitas das informações de performance dos algoritmos não estão disponíveis, como taxas de falsos positivos, falsos negativos, e troca de rótulos. Entretanto, apenas com a MOTA e o MOTP é possível se ter uma ideia do desempenho do RAMUK e RAMUK-DL.

Analisando a tabela 8, podemos dizer que ambos os algoritmos propostos mostraram-se competitivos com outras soluções da literatura. O RAMUK-DL obteve a quinta melhor acurácia entre os algoritmos comparados. As precisões de ambos ficaram um pouco aquém do melhor resultado, com uma diferença de pouco mais de 10%. Os resultados poderiam ser melhores se não houvessem tantos erros de correspondência. O RAMUK-DL mostrou uma ligeira superioridade com relação ao RAMUK em termos de precisão, acurácia e erros de correspondência. Isso mostra que o uso dos descritores foi útil na etapa de associação de dados que reduziu quase metade dos erros de correspondência. As heurísticas adotadas em RAMUK-DL também contribuíram para melhorar MOTP e MOTA, pois evitou por muitas vezes a junção de *blobs*.

Na tabela 9 fica evidente que os algoritmos desenvolvidos não concluíram o rastreamento com sucesso por conta do alto índice de

falsos negativos. Esses valores são decorrentes das não detecções de objetos pequenos (por conta da altura mínima), da sobreposição de objetos que ignoram as detecções desses objetos e de um modelo de plano de fundo ruim (já que as sequências são curtas e não é possível construir um modelo confiável em um tempo curto).



Figura 5.8 – Média dos modelos de plano de fundo da sequência ShopAssistant2cor

A figura 5.8 mostra um exemplo de uma média dos modelos de plano de fundo. É possível notar várias manchas no corredor, até mesmo uma pessoa parada, o que pode gerar dúvidas sobre a pertinência de um *pixel* ao modelo de *background*.

A diferença entre os resultados do RAMUK e RAMUK-DL considerando o CAVIAR são muito pequenas. Como existem longas sobreposições e uma detecção ruim, a heurística do RAMUK-DL não é aproveitada, pois não são encontrados pontos-chave válidos para se aplicar o FREAK.

6 CONSIDERAÇÕES FINAIS

6.1 CONCLUSÕES

Essa dissertação abordou um dos temas mais populares da área de visão computacional, que é o rastreamento de objetos. A relevância desse tema decorre do grande número de aplicações nas quais o rastreamento pode ser utilizado.

Seguindo a tendência em usar algoritmos baseados na extração de plano de fundo (ARROYO *et al.*, 2015; V, SPANG, 2014; WU; LIM; YANG, 2015), foram propostos dois algoritmos: RAMUK e RAMUK-DL. Ambos realizam a segmentação da imagem através do algoritmo ViBe e usam o Filtro de Kalman para estimar estados futuros, mas o último faz uso de heurísticas baseadas em descritores FREAK para ajudar na etapa de associação de dados.

Os resultados do RAMUK e do RAMUK-DL mostram que é possível obter bons resultados apenas com uso do FK associado a uma heurística para tratamento de oclusões. A versão que faz uso dos descritores locais contribuiu positivamente para que o algoritmo aumentasse sua acurácia e precisão, pois além de ajudar na localização de objetos durante a sobreposição deles, ele reduz as trocas de rótulos.

Embora tenham sido comparados apenas com vídeos cujos alvos são pessoas, os algoritmos propostos podem ser utilizados para detectar qualquer tipo de objeto de interesse. Essa versatilidade pode ser uma desvantagem ao se comparar com resultados de algoritmos baseados em modelos de aparência, que por serem especialistas, podem gerar resultados melhores.

Quanto ao rastreamento de pessoas, os algoritmos mostraram resultados competitivos com outros do estado da arte, com resultados superiores a muitos deles para a sequência S2L1. Entretanto essa realidade é diferente para o *dataset* CAVIAR. A curta duração dos vídeos impactou na má formação dos modelos de plano de fundo, o que traz problemas como o aparecimento de fantasmas e *clutter*. Além disso, o posicionamento de câmera adotado nas sequências do CAVIAR favorece longas oclusões de objetos, fazendo com que muitas detecções fossem ignoradas pelos algoritmos. Outros fatores como reflexos de objetos em vidros e pisos também contribuíram para o mau desempenho dos algoritmos sobre o CAVIAR.

Pode-se dizer que é possível utilizar os algoritmos propostos em cenas de videovigilância de ambientes externos, onde a câmera se encontra mais distante dos alvos. Já o uso desses algoritmos em

ambientes internos deve ser mais investigado, realizando testes em outros vídeos para averiguar o desempenho dos algoritmos propostos em ambientes fechados.

Dessa forma, pode-se dizer que o objetivo principal foi atingido com sucesso, pois foram propostos não apenas um, mas dois algoritmos de rastreamento. Os objetivos específicos também foram plenamente alcançados, exceto o objetivo de performance que foi parcialmente atendido, pois o desempenho dos algoritmos não foram competitivos sob o *dataset* CAVIAR.

6.2 TRABALHOS FUTUROS

Uma das dificuldades em estudar o rastreamento de múltiplos objetos é encontrar estudos comparativos que usam a mesma base de dados e métricas de avaliação. Além disso, muitos *datasets* não disponibilizam o gabarito das sequências, tornando muito difícil a avaliação dos resultados. Com isso, uma forma interessante de explorar o tema de rastreamento de objetos no DAS seria desenvolver uma base de dados própria, separando vídeos por aplicações, contemplando os desafios que acercam essas aplicações. Para tornar esse *dataset* relevante, deve-se disponibilizar os vídeos e seus respectivos gabaritos para que pesquisadores da área possam usá-los em seus trabalhos acadêmicos. Ademais, até que esse *dataset* ganhe importância na comunidade de visão computacional, deve-se implementar e testar algoritmos clássicos e atuais nessa base de dados, gerando artigos que ajudaram a divulgar o *dataset* do DAS.

Este trabalho pode ser aprimorado em diversos aspectos, principalmente com relação ao tempo de execução dos algoritmos. O tempo de execução pode ser determinante na escolha de uma técnica para realizar o rastreamento, entretanto esse item não foi incluído nos objetivos deste estudo. A execução de ambos os algoritmos é bem lenta, sendo a etapa de segmentação a mais lenta delas. Os algoritmos propostos podem ser aprimorados ao se adotar uma linguagem de programação mais leve que o MATLAB[®] como C++, mas o que realmente vai reduzir o tempo de execução dos algoritmos é utilizar processamento paralelo em múltiplos núcleos como CUDA que executam o processamento na GPU.

Pode-se estudar outros métodos para predição de estados futuros como o filtro de partículas. O filtro de partículas pode gerar resultados melhores, pois geralmente é usado em situações onde a modelagem de um sistema é complexa, que é o caso do rastreamento de pessoas. Pode-

se explorar também outros algoritmos de extração de *background* como o ViBe+, KDE e PBAS. Segundo Goyette *et al.* (2012) estes algoritmos apresentaram uma segmentação melhor que o ViBe e poderiam apurar a detecção de objetos.

Outra vertente de estudo com base neste trabalho seria investigar o uso de outros descritores locais binários como o ORB, SURF e BRISK. Pode ser que alguns desses descritores apresente resultados melhores que os obtidos com o FREAK.

REFERÊNCIAS

- ALAHÍ, A.; ORTIZ, R.; VANDERGHEYNST, P. **FREAK: Fast Retina Keypoint**. 2012 IEEE Conference on Computer Vision and Pattern Recognition. **Anais...IEEE**, jun. 2012. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6247715>>
- ALBERNAZ, P. D. F. **Filtragem multi-alvo aplicada ao rastreamento de alvos manobrantés e alvos balísticos**. 95p. Dissertação de Mestrado –Universidade Estadual de Campinas, Campinas, 2016.
- ANDRADE, F. S. P. *et al.* Fusion of Local and Global Descriptors for Content-Based Image and Video Retrieval. In: Campinas, São Paulo: [s.n.]. p. 845–853.
- ARNAUT, R. D.; COSTA, A. H. R. Rastreamento de jogadores de futebol em seqüências de imagens. **Revista TECCEN – Edição Especial**, v. 2, n. 1, p. 44–53, 2009.
- ARROYO, R. *et al.* Expert video-surveillance system for real-time detection of suspicious behaviors in shopping malls. **Expert Systems with Applications**, v. 42, n. 21, p. 7991–8005, 2015.
- BAE, S. H.; YOON, K. J. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. **Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition**, p. 1218–1225, 2014.
- BARNICH, O.; VAN DROOGENBROECK, M. **ViBE: A powerful random technique to estimate the *background* in video sequences**. 2009 IEEE International Conference on Acoustics, Speech and Signal Processing. **Anais...IEEE**, abr. 2009. Disponível em: <<http://ieeexplore.ieee.org/document/4959741/>>
- BARNICH, O.; VAN DROOGENBROECK, M. ViBe: A Universal *Background* Subtraction Algorithm for Video Sequences. **IEEE Transactions on Image Processing**, v. 20, n. 6, p. 1709–1724, jun. 2011.
- BAY, H. *et al.* Speeded-Up Robust Features (SURF). **Computer Vision and Image Understanding**, v. 110, n. 3, p. 346–359, jun. 2008.

BERCLAZ, J. *et al.* Multiple object tracking using k-shortest paths optimization. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 33, n. 9, p. 1806–1819, 2011.

BERNARDIN, K.; ELBS, A.; STIEFELHAGEN, R. Multiple object tracking performance metrics and evaluation in a smart room environment. **Sixth IEEE International Workshop on Visual Surveillance, in conjunction with ECCV**, v. 90, p. 91, 2006.

BERNARDIN, K.; STIEFELHAGEN, R. Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. **EURASIP Journal on Image and Video Processing**, v. 2008, p. 1–10, 2008.

CANNONS, K. **A Review of Visual Tracking CSE-2008-07**. Toronto, ON, Canada: [s.n.]. Disponível em:
<<http://www.cse.yorku.ca/techreports/2008/CSE-2008-07.pdf>>.

CASAGRANDE, J. H. B. **Detecção de Movimento Anormal em Videovigilância Baseada em Rastreamento e Agrupamentos Uniformes Ótimos**. 174p. Tese de Doutorado – DAS, Universidade Federal de Santa Catarina, 2015.

CHALLA, S. *et al.* **Fundamentals of Object Tracking**. 1. ed. Cambridge: Cambridge University Press, 2011. v. 1

CHOI, J.; MOON, D.; YOO, J. Robust Multi-person Tracking for Real-Time Intelligent Video Surveillance. **ETRI Journal**, v. 37, n. 3, p. 551–561, 2015.

COMUNELLO, E. **Visão Computacional**. Disponível em:
<<http://www.lapix.ufsc.br/ensino/visao-computacional>>.

COSTA, B. A. D. DA. **Segmentação, Rastreamento de Objetos e Detecção de Eventos Primitivos com Aplicação no Monitoramento Automático de Ações Humanas em Vídeo**. 129p. Dissertação de Mestrado – Universidade Federal de Campina Grande, Campina Grande, 2008.

DALAL, N. **Finding People in Images and Videos**. 149p. Tese de Doutorado – Institut National Polytechnique de Grenoble-INPG., Saint Ismier, França, 2006.

DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. **Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005**, v. I, p. 886–893, 2005.

DEORI, B.; THOUNAOJAM, D. M. a Survey on Moving Object Tracking in Video. **International Journal on Information Theory (IJIT)**, v. 3, n. 3, p. 31–46, 2014.

DUBUISSON, S.; GONZALES, C. A survey of *datasets* for visual tracking. **Machine Vision and Applications**, v. 27, n. 1, p. 23–52, 25 jan. 2016.

ELAFI, I.; JEDRA, M.; ZAHID, N. Unsupervised detection and tracking of moving objects for video surveillance applications. **Pattern Recognition Letters**, v. 84, p. 70–77, dez. 2016.

FAN, C.-S. *et al.* A Survey of Intelligent Video Surveillance Systems: History, Applications and Future. **Intelligent Systems and Applications**, v. 274, n. Intelligent Systems and Applications, p. 1479–1488, 2015.

GONZÁLES, G. L. G. **Aplicação da Técnica SIFT para Determinação de Campos de Deformações de Materiais usando Visão Computacional**. 109p. Dissertação de Mestrado – PUC-Rio, 2010.

GOYETTE, N. *et al.* changedetection.net: A new change detection benchmark *dataset*. **IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops**, p. 1–8, 2012.

HAN, G.; WANG, J.; CAI, X. *Background* Subtraction Based on Three-Dimensional Discrete Wavelet Transform. **Sensors**, v. 16, n. 4, p. 456, 30 mar. 2016.

HARITAOGLU, I.; HARWOOD, D.; DAVIS, L. S. W4: Real-time surveillance of people and their activities. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 22, n. 8, p. 809–830, 2000.

HOFMANN, M.; TIEFENBACHER, P.; RIGOLL, G. **Background segmentation with feedback: The Pixel-Based Adaptive Segmenter**. 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. **Anais...IEEE**, jun. 2012. Disponível em: <<http://ieeexplore.ieee.org/document/6238925/>>

ILEA, D. E.; WHELAN, P. F. Image segmentation based on the integration of colour texture descriptors - A review. **Pattern Recognition**, v. 44, n. 10–11, p. 2479–2501, 2011.

KALAL, Z.; MIKOLAJCZYK, K.; MATAS, J. Tracking-Learning-Detection. **IEEE transactions on pattern analysis and machine intelligence**, v. 34, n. 7, p. 1409–1422, 2011.

KAMPF, C. **Liberdade monitorada: a crescente vigilância eletrônica prejudica a privacidade?** Disponível em: <http://comciencia.scielo.br/scielo.php?script=sci_arttext&pid=S1519-76542013000200005&lng=pt&nrm=iso>. Acesso em: 2 fev. 2017.

KHAN, Z.; GU, I.; BACKHOUSE, A. Robust visual object tracking using multi-mode anisotropic mean shift and particle filters. **Circuits and Systems for ...**, v. 21, n. 1, p. 74–87, 2011.

KIM, C. *et al.* **Multiple Hypothesis Tracking Revisited**. 2015 IEEE International Conference on Computer Vision (ICCV). **Anais...IEEE**, dez. 2015. Disponível em: <<http://ieeexplore.ieee.org/document/7410890/>>

KO, T. **A survey on behavior analysis in video surveillance for homeland security applications**. 2008 37th IEEE Applied Imagery Pattern Recognition Workshop. **Anais...Washington, DC: IEEE**, 2008. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4906450>>

KRYJAK, T.; GORGON, M. Real-time implementation of the ViBe foreground object segmentation algorithm. **Computer Science and Information Systems (FedCSIS), 2013 Federated Conference on**, n. 1, p. 591–596, 2013.

LEAL-TAIXÉ, L. *et al.* MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking. p. 1–15, 2015.

LEUTENEGGER, S.; CHLI, M.; SIEGWART, R. Y. **BRISK: Binary Robust invariant scalable keypoints**. 2011 International Conference on Computer Vision. **Anais...IEEE**, nov. 2011. Disponível em: <<http://ieeexplore.ieee.org/document/6126542/>>

LI, X. *et al.* A survey of appearance models in visual object tracking. **ACM Transactions on Intelligent Systems and Technology**, v. 4, n. 4, p. 1–48, 1 set. 2013.

LLOYD, K. *et al.* Detecting Violent and Abnormal Crowd activity using Temporal Analysis of Grey Level Co-occurrence Matrix (GLCM) Based Texture Measures. **Machine Vision and Applications**, v. 28, n. 3–4, p. 361–371, 17 maio 2016.

LOWE, D. G. Object recognition from local scale-invariant features. **Proceedings of the Seventh IEEE International Conference on Computer Vision**, p. 1150–1157 vol.2, 1999.

_____. Distinctive image features from scale invariant keypoints. **International Journal of Computer Vision**, v. 60, p. 91–11020042, 2004.

MAGGIO, E.; CAVALLARO, A. **Video Tracking**. Chichester, UK: John Wiley & Sons, Ltd, 2011.

MAIR, E. *et al.* Adaptive and Generic Corner Detection Based on the Accelerated Segment Test. In: **European Conference on Computer Vision**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 183–196.

MARESCA, M. E.; PETROSINO, A. MATRIOSKA: A Multi-level Approach to Fast Tracking by Learning. In: PETROSINO, A. (Ed.). **Image Analysis and Processing – ICIAP 2013**. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. v. 8157p. 419–428.

MARTINS, J. G. *et al.* Uso de descritores de textura extraídos de GLCM para o reconhecimento de padrões em diferentes domínios de aplicação. **XXXVII Conferencia Latinoamericana de Informática**, p. 637–652, 2011.

MILAN, A.; ROTH, S.; SCHINDLER, K. Continuous Energy Minimization for Multitarget Tracking. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 36, n. 1, p. 58–72, jan. 2014.

MONTANARI, R. **Detecção e classificação de objetos em imagens para rastreamento de veículos**. 77p. Dissertação de Mestrado–Universidade de São Paulo, São Paulo, 2015.

MUSCI, M. *et al.* An evaluation of texture descriptors based on local binary patterns for classifications of remote sensing images. **Boletim de Ciências Geodésicas**, v. 17, n. 4, p. 549–570, 2011.

NARAYANA, M.; HAVERKAMP, D. **A Bayesian algorithm for tracking multiple moving objects in outdoor surveillance video**. 2007 IEEE Conference on Computer Vision and Pattern Recognition. **Anais...Mineapolis, MN: IEEE**, jun. 2007. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4270444>>

NEBEHAY, G.; PFLUGFELDER, R. **Consensus-based matching and tracking of keypoints for object tracking**. IEEE Winter Conference on Applications of Computer Vision. **Anais...IEEE**, mar. 2014. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6836013>>

NIETO, M. **Public Video Surveillance: Is it an Effective Crime Prevention Tool?** Sacramento, CA, USA: [s.n.]. Disponível em: <<https://www.library.ca.gov/CRB/97/05/crb97-005.pdf>>. Acesso em: 14 mar. 2016.

NUNES, M. C. DE P. T. P. **Videovigilância da Prevenção à Repressão Questões de violação da privacidade e valia probatória**. 46p. Mestrado Forense – Universidade Católica Portuguesa, Lisboa, Portugal, 2011.

OJHA, S.; SAKHARE, S. Image processing techniques for object tracking in video surveillance- A survey. **2015 International Conference on Pervasive Computing (ICPC)**, v. 0, n. c, p. 1–6, 2015.

PANG, S. *et al.* Robust Object Tracking Based on ORB Features and Particle Filter. **Journal of Information and Computational Science**, v. 10, n. 6, p. 1641–1649, 10 abr. 2013.

PENATTI, O. A. B.; DA SILVA TORRES, R. **Color descriptors for Web image retrieval: A comparative study**. Proceedings - 21st Brazilian Symposium on Computer Graphics and Image Processing, SIBGRAPI 2008. **Anais...**2008

PIVA, S.; SPIRITO, M.; REGAZZONI, C. S. The Use of Color Features in Automatic Video Surveillance Systems. In: **Color Image Processing: Emerging Applications**. [s.l.] CRC Press, 2006. p. 549–566.

RAJPOOT, Q. M.; JENSEN, C. D. Video surveillance: Privacy issues and legal compliance. In: **Promoting Social Change and Democracy through Information Technology**. Hershey, PA: IGI Global, 2015. p. 69–92.

RÄTY, T. D. Survey on contemporary remote surveillance systems for public safety. **IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews**, v. 40, n. 5, p. 493–515, 2010.

ROSHAN ZAMIR, A.; DEGHAN, A.; SHAH, M. GMCP-Tracker: Global Multi-object Tracking Using Generalized Minimum Clique Graphs. In: **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**. [s.l.: s.n.]. v. 7573 LNCSp. 343–356.

ROSTEN, E.; PORTER, R.; DRUMMOND, T. Faster and better: A machine learning approach to corner detection. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 32, n. 1, p. 105–119, 2010.

ROYO, C. V. **Image-Based Query by Example Using MPEG-7 Visual Descriptors**. 98p. Dissertação de Mestrado – Universitat Politècnica de Catalunya, Barcelona, Espanha, 2010.

SANTOS, R. J. B. C. *et al.* **Rastreamento de Objetos Usando Template Matching e Variação de Iluminação**. 14th Symposium on Virtual and Augmented Reality. **Anais...**2012. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/svr/2012/0018.pdf>>

SCHAEFFER, C. A Comparison of Keypoint Descriptors in the Context of Pedestrian Detection: FREAK vs. SURF vs. BRISK. **Cité en p**, v. 12, 2013.

SHAH, M. **Histograms of Oriented Gradients (HOG)**. Disponível em: <https://www.youtube.com/watch?v=0Zib1YEE4LU&index=30&list=PLQ1nfh0tq7KOReZp8QtxLWO_mSM7yfHT9>. Acesso em: 30 jan. 2017.

SILVA, A. T. DA. **Descritores de Imagem**. 26 slides. Apresentação em PowerPoint. Joinville, 2016.

SOUZA, M. L. DE. **Rastreamento de Aeronaves em Manobras via Combinação de Descarte e Fusão de Hipóteses**. 91p. Dissertação de Mestrado – Instituto Militar de Engenharia, Rio de Janeiro, 2015.

TAKALA, V.; PIETIKAINEN, M. Multi-object tracking using color, texture and motion. **IEEE Conference on Computer Vision and Pattern Recognition**, p. 1–7, 2007.

V, SPANG, H. **Object Tracking Using Local Binary Descriptors**. 96p. Dissertação de Mestrado – Rochester Institute of Technology, Rochester, EUA, 2014.

VAN DE SANDE, K. E. A.; GEVERS, T.; SNOEK, C. G. M. Evaluating Color Descriptors for Object and Scene Recognition. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 32, n. 9, p. 1582–1596, set. 2010.

VAN DROOGENBROECK, M.; PAQUOT, O. *Background subtraction: Experiments and improvements for ViBe*. **IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops**, p. 32–37, 2012.

WELSH, B. C.; FARRINGTON, D. P. Effects of Closed-Circuit Television on Crime. **The ANNALS of the American Academy of Political and Social Science**, v. 587, n. May 2003, p. 110–135, 2003.

WU, Y.; LIM, J.; YANG, M.-H. **Online Object Tracking: A Benchmark**. 2013 IEEE Conference on Computer Vision and Pattern Recognition. **Anais...**Portland, OR: IEEE, jun. 2013. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6619156>>

WU, Y.; LIM, J.; YANG, M. H. Object tracking benchmark. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 37, n. 9, p. 1834–1848, 2015.

XIANG, Y.; ALAHI, A.; SAVARESE, S. Learning to track: Online multi-object tracking by decision making. **Proceedings of the IEEE International Conference on Computer Vision**, v. 11–18–Dece, p. 4705–4713, 2016.

XIN XU *et al.* **Human behavior understanding for video surveillance: Recent advance**. 2010 IEEE International Conference on Systems, Man and Cybernetics. **Anais...**Istanbul: IEEE, out. 2010. Disponível em:
<<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5641773>>

YILMAZ, A.; JAVED, O.; SHAH, M. Object tracking. **ACM Computing Surveys**, v. 38, n. 4, p. 13–es, 2006.

YILMAZ, A.; XIN LI; SHAH, M. Contour-based object tracking with occlusion handling in video acquired using mobile cameras. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 26, n. 11, p. 1531–1536, nov. 2004.

ZHANG, L.; VAN DER MAATEN, L. **Structure Preserving Object Tracking**. 2013 IEEE Conference on Computer Vision and Pattern Recognition. **Anais...**Portland, Or: IEEE, jun. 2013. Disponível em:
<<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6619084>>

ZHONG, W.; LU, H.; YANG, M.-H. **Robust object tracking via sparsity-based collaborative model**. 2012 IEEE Conference on Computer Vision and Pattern Recognition. **Anais...**Providence, RI: IEEE, jun. 2012. Disponível em:
<<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6247882>>

ZHOU, H.; YUAN, Y.; SHI, C. Object tracking using SIFT features and mean shift. **Computer Vision and Image Understanding**, v. 113, n. 3, p. 345–352, 2009.

APÊNDICE I – Resultados do teste de parâmetros para o FK

1. Multiple Object Tracking Accuracy – MOTA (%)

DATASET			$\sigma=1$	$\sigma=5$	$\sigma=10$	$\sigma=15$	$\sigma=20$	$\sigma=30$	
Seqüência									
PETS2009 S2L1	$u=0$	$\omega=0,5$	86,59	86,90	84,88	83,17	80,55	73,14	
		$\omega=1$	85,55	87,69	86,90	85,46	84,88	83,17	
		MÉDIA	80,44	85,55	86,59	87,39	87,62	87,40	
		84,65%	$\omega=10$	78,54	85,63	85,55	86,17	86,59	87,39
		$\omega=15$	76,80	84,30	86,20	85,55	86,25	86,59	
	$u=0,01$	$\omega=0,5$	86,56	86,73	84,64	83,50	80,74	73,06	
		$\omega=1$	85,64	87,64	86,73	85,51	84,64	83,50	
		MÉDIA	80,70	85,64	86,56	87,13	87,62	87,40	
		84,64%	$\omega=10$	78,58	85,58	85,64	86,15	86,56	87,13
		$\omega=15$	76,75	84,35	86,03	85,64	86,25	86,56	
	$u=0,1$	$\omega=0,5$	86,61	86,80	85,14	82,62	79,12	71,28	
		$\omega=1$	86,08	87,60	86,80	85,88	85,14	82,62	
		MÉDIA	80,94	86,08	86,61	87,33	87,81	87,83	
		84,59%	$\omega=10$	78,62	85,51	86,08	86,45	86,61	87,33
		$\omega=15$	76,26	83,61	85,84	86,08	86,38	86,61	
	$u=1$	$\omega=0,5$	78,79	64,94	53,79	28,81	8,32	11,29	
		$\omega=1$	80,07	73,68	64,94	59,85	53,79	28,81	
		MÉDIA	77,97	80,07	78,79	76,96	75,42	72,17	
		67,02%	$\omega=10$	75,83	79,25	80,07	79,57	78,79	76,96
		$\omega=15$	74,76	78,86	79,84	80,07	79,45	78,79	
CAVIAR 2006 Valores médios considerando os vídeos: EnterExitCrossingPaths1cor EnterExitCrossingPaths2cor OneLeaveShopReenter2cor OneStopNoEnter1cor	$u=0$	$\omega=0,5$	43,96	43,44	44,13	44,19	44,49	44,18	
		$\omega=1$	43,76	43,42	43,44	43,56	44,13	44,19	
		MÉDIA	44,64	43,85	43,48	43,40	43,33	43,48	
		43,88%	$\omega=10$	44,70	43,57	43,85	43,40	43,41	43,40
		$\omega=15$	45,22	44,82	43,74	43,85	43,86	43,54	
	$u=0,01$	$\omega=0,5$	43,96	43,44	44,13	44,12	44,00	43,89	
		$\omega=1$	43,76	43,42	43,44	43,49	44,13	44,12	
		MÉDIA	44,64	43,85	43,48	43,40	43,33	43,48	
		43,84%	$\omega=10$	44,70	43,57	43,85	43,40	43,35	43,40
		$\omega=15$	45,22	44,82	43,51	43,85	43,81	43,54	
	$u=0,1$	$\omega=0,5$	43,65	42,54	42,00	41,95	42,22	42,03	
		$\omega=1$	43,68	43,28	42,54	42,45	42,00	41,95	
		MÉDIA	44,77	43,65	43,33	43,20	43,28	42,62	
		43,23%	$\omega=10$	44,54	43,52	43,59	43,34	43,26	43,20
		$\omega=15$	45,18	44,64	43,76	43,65	43,59	43,39	
$u=1$	$\omega=0,5$	40,53	36,70	27,92	23,77	6,62	3,99		
	$\omega=1$	41,34	39,34	36,70	33,72	27,92	23,77		
	MÉDIA	42,96	41,14	40,02	39,50	39,99	38,52		
	35,95%	$\omega=10$	42,64	41,22	40,70	40,61	39,89	39,86	
	$\omega=15$	42,92	42,42	41,60	41,14	40,77	40,40		

2. Multiple Object Tracking Precision – MOTP (%)

DATSET			$\sigma=1$	$\sigma=5$	$\sigma=10$	$\sigma=15$	$\sigma=20$	$\sigma=30$	
Sequência									
PETS2009 S2L1	$u=0$ MÉDIA 62,89%	$\omega=0,5$	65,39	62,92	59,05	55,68	53,12	48,88	
		$\omega=1$	65,23	64,66	62,92	60,82	59,05	55,68	
		$\omega=5$	64,57	65,23	65,39	65,27	64,94	64,21	
		$\omega=10$	64,84	65,64	65,23	65,39	65,39	65,27	
		$\omega=15$	64,92	65,09	65,75	65,23	65,40	65,39	
	$u=0,01$ MÉDIA 62,86%	$\omega=0,5$	65,38	62,94	58,78	55,62	52,93	48,73	
		$\omega=1$	65,26	64,67	62,94	60,78	58,78	55,62	
		$\omega=5$	64,84	65,26	65,38	65,24	64,95	64,22	
		$\omega=10$	64,82	65,68	65,26	65,39	65,38	65,24	
		$\omega=15$	64,93	65,07	65,79	65,26	65,38	65,38	
	$u=0,1$ MÉDIA 62,14%	$\omega=0,5$	65,11	62,28	57,24	53,11	49,30	43,76	
		$\omega=1$	65,42	64,21	62,28	59,85	57,24	53,11	
		$\omega=5$	64,62	65,42	65,11	64,96	64,71	63,97	
		$\omega=10$	64,92	65,35	65,42	65,42	65,11	64,96	
		$\omega=15$	64,68	64,89	65,73	65,42	65,35	65,11	
	$u=1$ MÉDIA 52,19%	$\omega=0,5$	61,51	44,59	30,08	26,69	29,10	35,80	
		$\omega=1$	62,99	55,66	44,59	35,98	30,08	26,69	
		$\omega=5$	63,94	62,99	61,51	59,83	57,88	53,15	
		$\omega=10$	63,97	63,78	62,99	62,22	61,51	59,83	
		$\omega=15$	64,00	63,83	63,44	62,99	62,57	61,51	
	CAVIAR 2006 Valores médios considerando os vídeos: EnterExitCrossingPaths1cor EnterExitCrossingPaths2cor OneLeaveShopReenter2cor OneStopNoEnter1cor	$u=0$ MÉDIA 63,07%	$\omega=0,5$	63,46	63,25	62,78	62,55	62,53	62,40
			$\omega=1$	63,62	63,41	63,25	63,05	62,78	62,55
			$\omega=5$	62,02	63,54	63,56	63,52	63,51	63,30
			$\omega=10$	61,63	63,44	63,52	63,61	63,58	63,51
$\omega=15$			61,93	61,85	63,48	63,55	63,48	63,56	
$u=0,01$ MÉDIA 63,13%		$\omega=0,5$	63,48	63,29	62,86	62,74	62,88	62,73	
		$\omega=1$	63,62	63,45	63,29	63,09	62,86	62,74	
		$\omega=5$	62,01	63,54	63,58	63,53	63,52	63,34	
		$\omega=10$	61,62	63,45	63,54	63,60	63,59	63,52	
		$\omega=15$	61,94	61,86	63,47	63,55	63,52	63,57	
$u=0,1$ MÉDIA 62,58%		$\omega=0,5$	63,57	63,14	62,51	60,93	58,74	54,03	
		$\omega=1$	63,67	63,70	63,14	62,78	62,51	60,93	
		$\omega=5$	61,95	63,66	63,67	63,79	63,62	63,36	
		$\omega=10$	61,67	63,31	63,60	63,64	63,69	63,73	
		$\omega=15$	62,04	61,82	63,29	63,67	63,62	63,64	
$u=1$ MÉDIA 52,65	$\omega=0,5$	62,28	46,28	29,39	24,79	23,74	35,38		
	$\omega=1$	62,69	57,32	46,28	35,32	29,39	24,79		
	$\omega=5$	61,40	62,62	61,77	60,16	58,48	54,56		
	$\omega=10$	61,55	62,90	62,23	61,93	61,38	60,48		
	$\omega=15$	61,51	61,52	62,61	62,66	62,20	62,00		

3. Taxa de Falsos Positivos – FP (%)

DATSET			$\sigma=1$	$\sigma=5$	$\sigma=10$	$\sigma=15$	$\sigma=20$	$\sigma=30$
Sequência								
PETS2009 S2L1	$u=0$ <i>MÉDIA</i> 4,60%	$\omega=0,5$	3,69	3,00	4,12	4,89	6,15	10,17
		$\omega=1$	4,56	2,60	3,00	3,90	4,12	4,89
		$\omega=5$	7,28	4,56	3,69	3,08	2,74	2,79
		$\omega=10$	8,40	4,11	4,56	4,09	3,69	3,08
		$\omega=15$	9,82	4,85	3,75	4,56	4,11	3,69
	$u=0,01$ <i>MÉDIA</i> 4,58%	$\omega=0,5$	3,69	3,07	4,20	4,65	6,08	10,10
		$\omega=1$	4,46	2,65	3,07	3,87	4,20	4,65
		$\omega=5$	7,17	4,46	3,69	3,20	2,74	2,79
		$\omega=10$	8,38	4,14	4,46	4,09	3,69	3,20
		$\omega=15$	9,84	4,83	3,78	4,46	4,11	3,69
	$u=0,1$ <i>MÉDIA</i> 4,57%	$\omega=0,5$	3,67	2,93	3,88	5,21	7,26	11,34
		$\omega=1$	4,21	2,67	2,93	3,40	3,88	5,21
		$\omega=5$	6,72	4,21	3,67	2,89	2,46	2,36
		$\omega=10$	8,31	4,20	4,21	3,81	3,67	2,89
		$\omega=15$	10,03	5,39	3,90	4,21	3,86	3,67
	$u=1$ <i>MÉDIA</i> 13,81%	$\omega=0,5$	7,32	15,02	21,06	33,53	44,42	46,22
		$\omega=1$	6,69	9,93	15,02	17,94	21,06	33,53
		$\omega=5$	8,09	6,69	7,32	8,29	9,10	10,95
		$\omega=10$	9,59	7,17	6,69	7,07	7,32	8,29
		$\omega=15$	10,56	7,52	6,84	6,69	7,11	7,32
CAVIAR 2006 Valores médios considerando os vídeos: EnterExitCrossingPaths1cor EnterExitCrossingPaths2cor OneLeaveShopReenter2cor OneStopNoEnter1cor	$u=0$ <i>MÉDIA</i> 15,32%	$\omega=0,5$	15,25	15,52	15,26	15,20	15,11	15,29
		$\omega=1$	15,26	15,53	15,52	15,37	15,26	15,20
		$\omega=5$	15,04	15,27	15,53	15,55	15,59	15,49
		$\omega=10$	14,98	15,51	15,27	15,54	15,57	15,55
		$\omega=15$	14,85	14,82	15,37	15,27	15,27	15,49
	$u=0,01$ <i>MÉDIA</i> 15,35%	$\omega=0,5$	15,25	15,52	15,26	15,24	15,39	15,45
		$\omega=1$	15,26	15,53	15,52	15,41	15,26	15,24
		$\omega=5$	15,04	15,27	15,53	15,55	15,59	15,49
		$\omega=10$	14,98	15,51	15,27	15,54	15,58	15,55
		$\omega=15$	14,85	14,82	15,56	15,27	15,30	15,49
	$u=0,1$ <i>MÉDIA</i> 15,65%	$\omega=0,5$	15,39	16,00	16,37	16,42	16,30	16,38
		$\omega=1$	15,29	15,57	16,00	16,04	16,37	16,42
		$\omega=5$	14,93	15,29	15,58	15,63	15,57	15,95
		$\omega=10$	15,07	15,51	15,39	15,55	15,62	15,63
		$\omega=15$	14,88	14,85	15,30	15,29	15,40	15,55
	$u=1$ <i>MÉDIA</i> 20,10%	$\omega=0,5$	17,05	19,54	24,90	27,77	37,53	43,13
		$\omega=1$	16,68	17,97	19,54	21,37	24,90	27,77
		$\omega=5$	15,75	16,77	17,39	17,76	17,48	18,52
		$\omega=10$	15,92	16,74	17,08	17,07	17,47	17,48
		$\omega=15$	15,80	16,07	16,55	16,77	16,99	17,13

4. Taxa de Falsos Negativos – FN (%)

DATSET			$\sigma=1$	$\sigma=5$	$\sigma=10$	$\sigma=15$	$\sigma=20$	$\sigma=30$
Sequência								
PETS2009 S2L1	$u=0$ MÉDIA 8,96%	$\omega=0,5$	8,10	8,57	9,29	10,09	11,32	14,22
		$\omega=1$	8,08	8,15	8,57	8,99	9,29	10,09
		$\omega=5$	9,95	8,08	8,10	8,01	8,08	8,24
		$\omega=10$	10,62	8,59	8,08	8,06	8,10	8,01
		$\omega=15$	10,81	8,92	8,43	8,08	7,94	8,10
	$u=0,01$ MÉDIA 8,99%	$\omega=0,5$	8,10	8,64	9,41	10,06	11,23	14,41
		$\omega=1$	8,08	8,15	8,64	8,97	9,41	10,06
		$\omega=5$	9,85	8,08	8,10	8,13	8,08	8,24
		$\omega=10$	10,60	8,62	8,08	8,08	8,10	8,13
		$\omega=15$	10,83	8,90	8,57	8,08	7,94	8,10
	$u=0,1$ MÉDIA 9,03%	$\omega=0,5$	8,08	8,69	9,25	10,37	11,44	14,76
		$\omega=1$	8,03	8,15	8,69	9,06	9,25	10,37
		$\omega=5$	9,97	8,03	8,08	8,17	8,17	8,22
		$\omega=10$	10,60	8,50	8,03	8,08	8,08	8,17
		$\omega=15$	11,02	8,99	8,62	8,03	8,10	8,08
	$u=1$ MÉDIA 15,61%	$\omega=0,5$	11,07	16,23	20,43	31,50	40,44	35,65
		$\omega=1$	10,48	13,36	16,23	18,09	20,43	31,50
		$\omega=5$	11,18	10,48	11,07	11,95	12,58	13,75
		$\omega=10$	11,46	10,88	10,48	10,65	11,07	11,95
		$\omega=15$	11,60	10,97	10,60	10,48	10,67	11,07
CAVIAR 2006 Valores médios considerando os vídeos: EnterExitCrossingPaths1cor EnterExitCrossingPaths2cor OneLeaveShopReenter2cor OneStopNoEnter1cor	$u=0$ MÉDIA 39,77%	$\omega=0,5$	39,79	40,03	39,64	39,64	39,47	39,60
		$\omega=1$	39,92	40,05	40,03	40,04	39,64	39,64
		$\omega=5$	39,25	39,82	39,99	40,05	40,07	40,03
		$\omega=10$	39,22	39,83	39,82	40,03	40,01	40,05
		$\omega=15$	38,75	39,23	39,83	39,82	39,84	39,97
	$u=0,01$ MÉDIA 39,79%	$\omega=0,5$	39,79	40,03	39,64	39,67	39,66	39,72
		$\omega=1$	39,92	40,05	40,03	40,06	39,64	39,67
		$\omega=5$	39,25	39,82	39,99	40,05	40,07	40,03
		$\omega=10$	39,22	39,83	39,82	40,03	40,06	40,05
		$\omega=15$	38,75	39,23	39,85	39,82	39,85	39,97
	$u=0,1$ MÉDIA 40,09%	$\omega=0,5$	39,96	40,46	40,64	40,68	40,48	40,55
		$\omega=1$	39,96	40,15	40,46	40,49	40,64	40,68
		$\omega=5$	39,22	39,99	40,08	40,16	40,15	40,46
		$\omega=10$	39,25	39,86	39,96	40,07	40,11	40,16
		$\omega=15$	38,82	39,35	39,83	39,99	39,98	40,06
	$u=1$ MÉDIA 42,09%	$\omega=0,5$	41,15	41,82	44,39	45,00	51,12	48,43
		$\omega=1$	40,68	41,36	41,82	42,61	44,39	45,00
		$\omega=5$	40,08	40,75	41,25	41,43	41,11	41,47
		$\omega=10$	40,01	40,71	40,92	41,05	41,29	41,26
		$\omega=15$	39,97	40,24	40,53	40,75	41,00	41,20

5. Trocas de Rótulos (erros de correspondência) – EC

DATSET		$\sigma=1$	$\sigma=5$	$\sigma=10$	$\sigma=15$	$\sigma=20$	$\sigma=30$	
PETS2009 S2L1		$\omega=0,5$	66	62	69	75	80	101
	$u=0$	$\omega=1$	75	63	62	67	69	75
	MÉDIA	$\omega=5$	97	75	66	62	63	63
	73	$\omega=10$	102	68	75	69	66	62
		$\omega=15$	109	79	66	75	70	66
		$\omega=0,5$	67	63	71	72	79	99
	$u=0,01$	$\omega=1$	75	63	63	67	71	72
	MÉDIA	$\omega=5$	95	75	67	63	63	63
	73	$\omega=10$	102	68	75	69	67	63
		$\omega=15$	109	79	66	75	70	67
		$\omega=0,5$	67	64	70	73	89	108
	$u=0,1$	$\omega=1$	69	64	64	67	70	73
	MÉDIA	$\omega=5$	98	69	67	65	63	64
	74	$\omega=10$	103	73	69	68	67	65
		$\omega=15$	114	83	67	69	68	67
		$\omega=0,5$	116	161	204	272	313	350
	$u=1$	$\omega=1$	113	125	161	176	204	272
	MÉDIA	$\omega=5$	114	113	116	115	119	130
	152	$\omega=10$	131	111	113	112	116	115
		$\omega=15$	130	109	112	113	114	116
CAVIAR 2006 Valores médios considerando os vídeos: EnterExitCrossingPaths1cor EnterExitCrossingPaths2cor OneLeaveShopReenter2cor OneStopNoEnter1cor		$\omega=0,5$	7	7	7	7	7	7
	$u=0$	$\omega=1$	7	7	7	7	7	7
	MÉDIA	$\omega=5$	8	7	7	7	7	7
	7	$\omega=10$	8	8	7	7	7	7
		$\omega=15$	9	8	7	7	7	7
		$\omega=0,5$	7	7	7	7	7	7
	$u=0,01$	$\omega=1$	7	7	7	7	7	7
	MÉDIA	$\omega=5$	8	7	7	7	7	7
	7	$\omega=10$	8	8	7	7	7	7
		$\omega=15$	9	8	8	7	7	7
		$\omega=0,5$	7	7	7	7	7	7
	$u=0,1$	$\omega=1$	7	7	7	7	7	7
	MÉDIA	$\omega=5$	8	7	7	7	7	7
	7	$\omega=10$	8	8	7	7	7	7
		$\omega=15$	8	8	8	7	7	7
		$\omega=0,5$	9	14	22	29	39	54
$u=1$	$\omega=1$	9	9	14	17	22	29	
MÉDIA	$\omega=5$	9	9	9	9	10	10	
14	$\omega=10$	11	9	9	9	9	10	
	$\omega=15$	10	9	9	9	8	9	

APÊNDICE II - Resultados para o teste de altura mínima

DATASETS	MEDIDAS (%)	PARÂMETRO ALTURA MÍNIMA (pixels)				
		10	20	30	40	50
PETS-2009 S2L1	MOTA	79,08	83,98	87,89	88,67	91,75
	MOTA	80,05	83,97	87,11	90,55	92,64
	MOTA	81,57	85,14	86,65	90,83	90,99
	MOTA	79,93	82,13	86,30	88,39	90,12
	MOTA	79,30	85,03	85,30	90,03	90,01
	MÉDIA MOTA	79,99	84,05	86,65	89,69	91,10
	MOTP	49,80	52,34	64,07	51,54	43,92
	MOTP	53,08	53,64	64,75	52,13	44,93
	MOTP	48,96	51,96	62,82	53,67	44,95
	MOTP	52,99	52,41	63,00	53,14	43,43
	MOTP	46,37	59,27	74,74	35,32	64,10
	MÉDIA MOTP	50,24	53,92	65,88	49,16	48,27
CAVIAR 2006 EnterExitCrossingPaths1cor	MOTA	49,80	52,34	64,07	51,54	43,92
	MOTA	53,08	53,64	64,75	52,13	44,93
	MOTA	48,96	51,96	62,82	53,67	44,95
	MOTA	52,99	52,41	63,00	53,14	43,43
	MOTA	46,37	59,27	74,74	35,32	64,10
	MOTP	44,66	50,26	47,84	45,10	41,83
	MOTP	45,19	46,22	47,83	40,99	49,95
	MOTP	45,66	49,84	47,21	40,34	40,98
	MOTP	41,67	50,25	48,02	39,61	50,07
	MOTP	43,56	47,78	51,52	39,93	47,47
		MÉDIA MOTP	43,56	47,78	51,52	39,93
CAVIAR 2006 EnterExitCrossingPaths2cor	MOTA	18,12	26,27	28,59	28,98	34,81
	MOTA	23,61	24,76	31,47	28,92	32,07
	MOTA	19,12	23,04	24,30	29,07	34,20
	MOTA	18,99	23,47	28,49	33,67	29,17
	MOTA	20,64	24,43	31,24	27,60	36,39
	MOTP	68,54	71,27	65,44	75,54	70,77
	MOTP	56,54	72,57	45,41	73,20	71,65
	MOTP	63,95	67,78	55,41	73,45	67,38
	MOTP	54,01	71,05	47,81	75,48	72,01
	MOTP	57,02	69,18	47,59	74,79	71,09
		MÉDIA MOTP	57,02	69,18	47,59	74,79
CAVIAR 2006 OneLeaveShopReenter2cor	MOTA	34,05	38,65	24,38	14,84	37,29
	MOTA	35,50	37,62	29,31	21,62	15,47
	MOTA	32,41	31,52	21,33	8,04	12,13
	MOTA	36,23	36,29	27,47	16,57	36,71
	MOTA	33,37	37,41	26,37	15,50	37,81
	MOTP	46,35	48,53	46,48	53,47	43,97
	MOTP	46,27	40,26	46,10	54,99	59,84
	MOTP	40,26	47,07	50,06	55,42	65,41
	MOTP	44,58	45,40	45,09	50,67	58,45
	MOTP	45,37	48,25	50,94	59,53	65,21
		MÉDIA MOTP	45,37	48,25	50,94	59,53
CAVIAR 2006 OneStopNoEnter1cor	MOTA	58,62	52,17	65,53	64,76	63,15
	MOTA	58,71	64,78	65,92	66,64	63,93
	MOTA	61,65	53,13	65,51	68,65	66,00
	MOTA	56,28	59,81	64,90	65,75	64,83
	MOTA	60,46	54,77	65,71	66,25	67,93
	MOTP	57,59	60,26	79,69	69,47	68,04
	MOTP	56,44	68,91	64,17	55,94	75,75
	MOTP	55,92	66,21	63,67	52,40	76,37
	MOTP	71,29	56,01	52,19	73,47	64,47
	MOTP	57,92	67,03	63,72	53,67	77,28
		MÉDIA MOTP	57,92	67,03	63,72	53,67
MÉDIAS PARA OS VÍDEOS DO CAVIAR 2006	MOTA	40,95	42,89	46,49	40,13	43,66
	MOTP	52,14	57,21	53,31	57,87	61,90
MÉDIAS DE MOTA E MOTP DE AMBOS DATASET		55,83	59,52	63,08	59,21	61,23

APÊNDICE III - Resultados para o teste de altura mínima

DATASETS	MEDIDAS (%)	PARÂMETRO ALTURA MÍNIMA (pixels)				
		10	20	30	40	50
PETS-2009 S2L1	MOTA	79,08	83,98	87,89	88,67	91,75
	MOTA	80,05	83,97	87,11	90,55	92,64
	MOTA	81,57	85,14	86,65	90,83	90,99
	MOTA	79,93	82,13	86,30	88,39	90,12
	MOTA	79,30	85,03	85,30	90,03	90,01
	MÉDIA MOTA	79,99	84,05	86,65	89,69	91,10
	MOTP	49,80	52,34	64,07	51,54	43,92
	MOTP	53,08	53,64	64,75	52,13	44,93
	MOTP	48,96	51,96	62,82	53,67	44,95
	MOTP	52,99	52,41	63,00	53,14	43,43
	MOTP	46,37	59,27	74,74	35,32	64,10
MÉDIA MOTP	50,24	53,92	65,88	49,16	48,27	
CAVIAR 2006 EnterExitCrossingPaths1cor	MOTA	49,80	52,34	64,07	51,54	43,92
	MOTA	53,08	53,64	64,75	52,13	44,93
	MOTA	48,96	51,96	62,82	53,67	44,95
	MOTA	52,99	52,41	63,00	53,14	43,43
	MOTA	46,37	59,27	74,74	35,32	64,10
	MOTP	44,66	50,26	47,84	45,10	41,83
	MOTP	45,19	46,22	47,83	40,99	49,95
	MOTP	45,66	49,84	47,21	40,34	40,98
	MOTP	41,67	50,25	48,02	39,61	50,07
	MOTP	43,56	47,78	51,52	39,93	47,47
	MÉDIA MOTP	43,56	47,78	51,52	39,93	47,47
CAVIAR 2006 EnterExitCrossingPaths2cor	MOTA	18,12	26,27	28,59	28,98	34,81
	MOTA	23,61	24,76	31,47	28,92	32,07
	MOTA	19,12	23,04	24,30	29,07	34,20
	MOTA	18,99	23,47	28,49	33,67	29,17
	MOTA	20,64	24,43	31,24	27,60	36,39
	MOTP	68,54	71,27	65,44	75,54	70,77
	MOTP	56,54	72,57	45,41	73,20	71,65
	MOTP	63,95	67,78	55,41	73,45	67,38
	MOTP	54,01	71,05	47,81	75,48	72,01
	MOTP	57,02	69,18	47,59	74,79	71,09
	MÉDIA MOTP	57,02	69,18	47,59	74,79	71,09
CAVIAR 2006 OneLeaveShopReenter2cor	MOTA	34,05	38,65	24,38	14,84	37,29
	MOTA	35,50	37,62	29,31	21,62	15,47
	MOTA	32,41	31,52	21,33	8,04	12,13
	MOTA	36,23	36,29	27,47	16,57	36,71
	MOTA	33,37	37,41	26,37	15,50	37,81
	MOTP	46,35	48,53	46,48	53,47	43,97
	MOTP	46,27	40,26	46,10	54,99	59,84
	MOTP	40,26	47,07	50,06	55,42	65,41
	MOTP	44,58	45,40	45,09	50,67	58,45
	MOTP	45,37	48,25	50,94	59,53	65,21
	MÉDIA MOTP	45,37	48,25	50,94	59,53	65,21
CAVIAR 2006 OneStopNoEnter1cor	MOTA	58,62	52,17	65,53	64,76	63,15
	MOTA	58,71	64,78	65,92	66,64	63,93
	MOTA	61,65	53,13	65,51	68,65	66,00
	MOTA	56,28	59,81	64,90	65,75	64,83
	MOTA	60,46	54,77	65,71	66,25	67,93
	MOTP	57,59	60,26	79,69	69,47	68,04
	MOTP	56,44	68,91	64,17	55,94	75,75
	MOTP	55,92	66,21	63,67	52,40	76,37
	MOTP	71,29	56,01	52,19	73,47	64,47
	MOTP	57,92	67,03	63,72	53,67	77,28
	MÉDIA MOTP	57,92	67,03	63,72	53,67	77,28
MÉDIAS PARA OS VÍDEOS DO CAVIAR 2006	MOTA	40,95	42,89	46,49	40,13	43,66
MOTP	52,14	57,21	53,31	57,87	61,90	
MÉDIAS DE MOTA E MOTP DE AMBOS DATASET	MOTA	55,83	59,52	63,08	59,21	61,23
MOTP	50,24	53,92	65,88	49,16	48,27	

APÊNDICE IV – Resultados para o teste de pena máxima

SEQUÊNCIAS	MEDIDAS (%)	PARÂMETRO PENA MÁXIMA		
		3	8	15
PETS-2009 S2L1	MOTA	86,30	88,98	84,09
	MOTA	86,45	90,68	85,19
	MOTA	87,11	87,82	85,46
	MOTA	87,49	88,16	86,31
	MOTA	86,15	87,19	84,13
	MEDIA MOTA	86,70	88,57	85,04
	MOTP	66,52	66,72	64,05
	MOTP	67,34	67,27	64,75
	MOTP	66,16	67,56	65,12
	MOTP	65,65	67,44	63,94
	MOTP	65,96	64,81	64,51
MEDIA MOTP	66,33	66,76	64,47	
CAVIAR 2006 EnterExitCrossingPaths1cor	MOTA	43,91	53,03	39,32
	MOTA	48,36	35,61	44,56
	MOTA	47,76	34,65	43,92
	MOTA	48,52	50,32	44,43
	MOTA	46,78	48,31	44,09
	MOTP	37,35	42,59	41,63
	MOTP	37,12	42,46	42,50
	MOTP	40,22	43,87	40,63
	MOTP	40,14	44,07	41,29
	MOTP	43,50	40,49	36,92
	MEDIA MOTP	40,14	42,46	41,29
CAVIAR 2006 EnterExitCrossingPaths2cor	MOTA	32,11	33,42	31,43
	MOTA	38,19	32,35	27,66
	MOTA	37,07	29,96	26,71
	MOTA	36,83	30,36	29,16
	MOTA	35,66	30,17	27,17
	MOTP	67,11	65,85	61,76
	MOTP	66,47	66,73	61,82
	MOTP	66,84	65,41	64,95
	MOTP	67,11	65,94	64,77
	MOTP	66,19	65,78	56,25
	MEDIA MOTP	66,19	65,78	56,25
CAVIAR 2006 OneLeaveShopReenter2cor	MOTA	37,98	38,89	35,91
	MOTA	11,43	15,62	7,17
	MOTA	10,34	13,86	8,81
	MOTA	43,71	43,64	39,43
	MOTA	37,60	37,79	33,74
	MOTP	51,50	58,59	49,25
	MOTP	51,70	58,68	49,58
	MOTP	54,42	56,58	46,96
	MOTP	54,19	57,14	47,74
	MOTP	51,77	51,14	42,70
	MEDIA MOTP	51,77	51,14	42,70
CAVIAR 2006 OneStopNoEnter1cor	MOTA	53,91	59,40	62,48
	MOTA	55,78	58,74	60,88
	MOTA	54,70	50,88	60,44
	MOTA	54,70	58,58	60,70
	MOTA	55,37	51,68	62,01
	MOTP	75,03	52,04	73,17
	MOTP	74,82	51,63	74,08
	MOTP	74,03	64,95	58,13
	MOTP	74,78	78,24	72,91
	MOTP	74,82	77,60	73,75
	MEDIA MOTP	74,82	77,60	73,75
MÉDIAS PARA OS VÍDEOS DO CAVIAR 2006	MOTA	41,54	40,36	39,50
MOTP	58,45	57,49	55,04	
MÉDIAS DAS MÉDIAS DE MOTA E MOTP		63,25	63,30	61,01

APÊNDICE V – Resultados de performance do RAMUK e RAMUK-DL

1. RAMUK

SEQUÊNCIA	MOTP (%)	MOTA (%)	FP (%)	FN (%)	EC	MT (%)	PT (%)	ML (%)
PETS-2009 S2L1	65,31	86,31	2,67	9,67	53	89,47	10,53	0,00
PETS-2009 S2L1	68,87	88,57	1,66	8,42	53	89,47	10,53	0,00
PETS-2009 S2L1	67,90	85,27	2,44	10,94	53	89,47	10,53	0,00
PETS-2009 S2L1	68,78	88,37	2,06	8,23	53	89,47	10,53	0,00
PETS-2009 S2L1	68,36	85,22	3,64	9,79	53	89,47	10,53	0,00
PETS-2009 S2L1	70,16	86,57	2,15	9,93	53	89,47	10,53	0,00
PETS-2009 S2L1	67,42	87,06	4,38	7,21	53	89,47	10,53	0,00
PETS-2009 S2L1	68,99	90,79	0,87	7,00	53	89,47	10,53	0,00
PETS-2009 S2L1	71,28	86,51	3,74	8,40	53	89,47	10,53	0,00
PETS-2009 S2L1	63,77	88,26	2,90	7,49	53	89,47	10,53	0,00
EnterExitCrossingPaths1cor	51,75	34,09	11,30	54,05	3	40,00	20,00	40,00
EnterExitCrossingPaths1cor	52,98	37,81	8,89	52,70	3	40,00	20,00	40,00
EnterExitCrossingPaths1cor	51,77	37,67	9,31	52,42	3	40,00	20,00	40,00
EnterExitCrossingPaths1cor	52,12	39,47	8,67	51,26	3	40,00	20,00	40,00
EnterExitCrossingPaths1cor	56,48	38,61	10,24	50,55	3	40,00	20,00	40,00
EnterExitCrossingPaths1cor	52,98	37,81	8,89	52,70	3	40,00	20,00	40,00
EnterExitCrossingPaths1cor	56,27	39,72	11,34	48,34	3	40,00	20,00	40,00
EnterExitCrossingPaths1cor	56,46	35,08	8,29	56,03	3	40,00	20,00	40,00
EnterExitCrossingPaths1cor	57,12	37,12	10,92	51,36	3	40,00	20,00	40,00
EnterExitCrossingPaths1cor	54,20	35,29	11,94	52,17	3	40,00	20,00	40,00
EnterExitCrossingPaths2cor	73,72	80,51	8,74	10,26	2	100,00	0,00	0,00
EnterExitCrossingPaths2cor	73,86	77,85	8,77	13,13	1	100,00	0,00	0,00
EnterExitCrossingPaths2cor	77,04	75,75	9,91	14,14	1	100,00	0,00	0,00
EnterExitCrossingPaths2cor	72,04	77,18	9,75	12,87	1	100,00	0,00	0,00
EnterExitCrossingPaths2cor	77,61	80,05	5,41	14,34	1	100,00	0,00	0,00
EnterExitCrossingPaths2cor	76,90	80,48	6,88	12,44	1	100,00	0,00	0,00
EnterExitCrossingPaths2cor	80,26	79,73	10,51	9,56	1	100,00	0,00	0,00
EnterExitCrossingPaths2cor	76,19	78,99	6,41	14,39	1	100,00	0,00	0,00
EnterExitCrossingPaths2cor	79,03	80,11	6,96	12,73	1	100,00	0,00	0,00
EnterExitCrossingPaths2cor	76,25	80,62	6,67	12,51	1	100,00	0,00	0,00
OneLeaveShop1cor	50,72	29,35	5,30	64,64	2	0,00	40,00	60,00
OneLeaveShop1cor	44,96	38,04	0,00	60,95	3	0,00	60,00	40,00
OneLeaveShop1cor	44,10	43,98	0,00	55,42	3	0,00	60,00	40,00
OneLeaveShop1cor	45,67	41,04	0,00	58,36	3	0,00	60,00	40,00
OneLeaveShop1cor	45,83	39,06	0,00	60,34	3	0,00	60,00	40,00
OneLeaveShop1cor	49,63	39,40	0,00	60,00	3	0,00	60,00	40,00
OneLeaveShop1cor	43,04	41,21	0,00	58,19	3	0,00	60,00	40,00
OneLeaveShop1cor	47,57	38,40	0,00	61,00	3	0,00	60,00	40,00
OneLeaveShop1cor	48,52	37,69	0,00	61,71	3	0,00	60,00	40,00
OneLeaveShop1cor	51,02	40,02	0,00	59,38	3	0,00	60,00	40,00
OneLeaveShopReenter2cor	58,88	15,17	15,90	68,17	5	16,67	16,67	66,67
OneLeaveShopReenter2cor	59,50	19,41	11,98	67,82	5	16,67	16,67	66,67
OneLeaveShopReenter2cor	58,47	23,20	11,62	64,18	5	16,67	16,67	66,67

OneLeaveShopReenter2cor	60,14	26,13	10,66	62,22	5	16,67	16,67	66,67
OneLeaveShopReenter2cor	57,51	26,79	6,29	65,92	5	16,67	16,67	66,67
OneLeaveShopReenter2cor	59,81	24,04	10,97	64,00	5	16,67	16,67	66,67
OneLeaveShopReenter2cor	65,78	26,30	7,10	65,60	5	16,67	16,67	66,67
OneLeaveShopReenter2cor	64,34	25,70	6,09	67,21	5	16,67	16,67	66,67
OneLeaveShopReenter2cor	65,32	26,94	9,35	62,71	5	16,67	16,67	66,67
OneLeaveShopReenter2cor	58,30	21,40	12,48	65,12	5	16,67	16,67	66,67
OneShopOneWait1cor	57,78	50,72	0,78	47,89	14	14,29	57,14	28,57
OneShopOneWait1cor	55,97	47,54	0,96	50,81	15	14,29	57,14	28,57
OneShopOneWait1cor	55,58	42,88	0,87	53,24	15	14,29	57,14	28,57
OneShopOneWait1cor	61,41	41,84	1,67	53,49	15	14,29	57,14	28,57
OneShopOneWait1cor	61,87	48,79	1,21	47,00	15	14,29	57,14	28,57
OneShopOneWait1cor	53,99	43,95	1,81	51,24	15	14,29	57,14	28,57
OneShopOneWait1cor	60,32	42,33	2,44	52,23	15	14,29	57,14	28,57
OneShopOneWait1cor	62,96	43,04	0,27	53,69	15	14,29	57,14	28,57
OneShopOneWait1cor	59,78	45,27	2,45	49,28	15	14,29	57,14	28,57
OneShopOneWait1cor	60,30	43,86	3,41	49,73	15	14,29	57,14	28,57
OneShopOneWait2cor	60,08	37,41	4,48	56,92	37	30,00	50,00	20,00
OneShopOneWait2cor	47,05	30,92	7,99	57,76	106	10,00	70,00	20,00
OneShopOneWait2cor	47,53	10,07	9,98	58,76	106	10,00	70,00	20,00
OneShopOneWait2cor	48,42	15,20	6,74	56,86	106	10,00	70,00	20,00
OneShopOneWait2cor	45,79	17,96	5,92	54,92	106	10,00	70,00	20,00
OneShopOneWait2cor	51,04	14,43	9,89	54,48	106	10,00	70,00	20,00
OneShopOneWait2cor	45,67	12,58	8,69	57,53	106	10,00	70,00	20,00
OneShopOneWait2cor	50,59	18,89	7,37	52,54	106	10,00	70,00	20,00
OneShopOneWait2cor	49,28	13,43	9,46	55,91	106	10,00	70,00	20,00
OneShopOneWait2cor	53,83	21,09	5,87	51,84	106	10,00	70,00	20,00
OneStopEnter1cor	77,47	56,15	2,41	41,44	0	33,33	50,00	16,67
OneStopEnter1cor	76,32	54,95	2,83	41,64	7	33,33	50,00	16,67
OneStopEnter1cor	78,09	58,85	1,86	37,89	7	33,33	50,00	16,67
OneStopEnter1cor	78,34	57,36	3,73	37,51	7	33,33	50,00	16,67
OneStopEnter1cor	74,69	65,10	2,81	30,68	7	33,33	50,00	16,67
OneStopEnter1cor	80,12	64,32	1,27	33,01	7	33,33	50,00	16,67
OneStopEnter1cor	80,84	63,42	4,32	30,85	7	33,33	50,00	16,67
OneStopEnter1cor	81,06	57,80	1,22	39,58	7	33,33	50,00	16,67
OneStopEnter1cor	79,80	54,90	4,30	39,40	7	33,33	50,00	16,67
OneStopEnter1cor	74,42	63,58	2,49	32,52	7	33,33	50,00	16,67
OneStopEnter1cor	74,84	63,20	2,08	33,32	7	33,33	50,00	16,67
OneStopEnter1cor	75,52	56,70	2,50	39,40	7	33,33	50,00	16,67
OneStopEnter2cor	76,76	75,58	9,86	14,36	1	100,00	0,00	0,00
OneStopEnter2cor	79,70	71,87	11,03	16,90	1	100,00	0,00	0,00
OneStopEnter2cor	75,56	72,19	10,53	17,08	1	100,00	0,00	0,00
OneStopEnter2cor	76,90	73,52	10,16	16,12	1	100,00	0,00	0,00
OneStopEnter2cor	76,08	73,44	12,04	14,32	1	100,00	0,00	0,00
OneStopEnter2cor	75,31	75,57	10,05	14,17	1	100,00	0,00	0,00
OneStopEnter2cor	76,65	76,67	10,99	12,14	1	100,00	0,00	0,00
OneStopEnter2cor	79,11	73,79	11,52	14,48	1	100,00	0,00	0,00
OneStopMoveEnter1cor	51,32	26,10	3,47	62,03	84	15,79	63,16	21,05
OneStopMoveEnter1cor	52,13	27,16	3,11	61,33	84	10,53	68,42	21,05
OneStopMoveEnter1cor	54,24	24,63	3,62	63,35	84	10,53	68,42	21,05

OneStopMoveEnter1cor	54,21	22,60	5,10	63,90	84	10,53	68,42	21,05
OneStopMoveEnter1cor	53,17	25,04	4,74	61,82	84	10,53	68,42	21,05
OneStopMoveEnter1cor	52,65	24,76	3,33	63,50	84	10,53	68,42	21,05
OneStopMoveEnter1cor	49,69	23,35	3,54	64,71	84	10,53	68,42	21,05
OneStopMoveEnter1cor	54,16	24,55	6,23	60,82	84	10,53	68,42	21,05
OneStopMoveEnter1cor	50,55	26,52	1,61	63,47	84	10,53	68,42	21,05
OneStopMoveEnter1cor	54,17	25,12	5,17	61,30	84	10,53	68,42	21,05
OneStopMoveEnter2cor	60,21	0,24	11,28	86,88	8	18,18	18,18	63,64
OneStopMoveEnter2cor	65,02	0,00	12,19	86,21	8	18,18	18,18	63,64
OneStopMoveEnter2cor	62,60	2,36	11,02	85,02	8	27,27	0,00	72,73
OneStopMoveEnter2cor	61,14	2,46	10,40	85,54	8	27,27	0,00	72,73
OneStopMoveEnter2cor	60,53	2,34	9,69	86,37	8	27,27	0,00	72,73
OneStopMoveEnter2cor	62,81	2,11	12,22	84,08	8	27,27	0,00	72,73
OneStopMoveEnter2cor	59,87	3,57	9,77	85,06	8	27,27	0,00	72,73
OneStopMoveEnter2cor	65,43	1,47	12,22	84,71	8	27,27	0,00	72,73
OneStopMoveEnter2cor	66,38	0,65	11,78	85,97	8	27,27	0,00	72,73
OneStopMoveEnter2cor	66,05	2,29	9,67	86,45	8	27,27	0,00	72,73
OneStopMoveNoEnter1cor	55,50	24,81	2,36	72,83	0	25,00	25,00	50,00
OneStopMoveNoEnter1cor	55,64	23,85	5,58	70,57	0	25,00	25,00	50,00
OneStopMoveNoEnter1cor	57,61	24,80	5,86	69,34	0	25,00	25,00	50,00
OneStopMoveNoEnter1cor	58,46	24,15	3,25	72,60	0	25,00	25,00	50,00
OneStopMoveNoEnter1cor	60,09	25,85	3,56	70,60	0	25,00	25,00	50,00
OneStopMoveNoEnter1cor	55,38	23,55	7,28	69,17	0	25,00	25,00	50,00
OneStopMoveNoEnter1cor	57,19	26,67	6,23	67,10	0	25,00	25,00	50,00
OneStopMoveNoEnter1cor	57,86	26,27	6,38	67,34	0	25,00	25,00	50,00
OneStopMoveNoEnter1cor	54,25	31,94	2,87	65,19	0	25,00	25,00	50,00
OneStopMoveNoEnter1cor	55,13	24,49	7,41	68,10	0	25,00	25,00	50,00
OneStopNoEnter1cor	77,82	57,13	1,54	41,11	2	66,67	0,00	33,33
OneStopNoEnter1cor	78,11	56,33	1,87	41,57	2	66,67	0,00	33,33
OneStopNoEnter1cor	82,61	55,23	2,70	41,67	2	66,67	0,00	33,33
OneStopNoEnter1cor	81,02	52,27	2,86	44,47	2	66,67	0,00	33,33
OneStopNoEnter1cor	78,49	60,92	1,55	37,13	2	66,67	0,00	33,33
OneStopNoEnter1cor	80,07	54,75	1,54	43,31	2	66,67	0,00	33,33
OneStopNoEnter1cor	78,88	54,24	2,04	43,33	2	66,67	0,00	33,33
OneStopNoEnter1cor	78,69	53,97	2,50	43,13	2	66,67	0,00	33,33
OneStopNoEnter1cor	84,81	56,83	3,64	39,13	2	66,67	0,00	33,33
OneStopNoEnter1cor	79,07	60,82	1,55	37,23	2	66,67	0,00	33,33
OneStopNoEnter2cor	75,98	77,74	1,36	20,49	9	50,00	50,00	0,00
OneStopNoEnter2cor	75,85	74,58	2,04	22,54	18	33,33	66,67	0,00
OneStopNoEnter2cor	77,16	73,39	2,50	20,51	18	33,33	66,67	0,00
OneStopNoEnter2cor	75,42	76,26	0,58	19,56	18	33,33	66,67	0,00
OneStopNoEnter2cor	74,02	70,99	1,89	23,52	18	33,33	66,67	0,00
OneStopNoEnter2cor	74,07	75,35	1,07	19,98	18	33,33	66,67	0,00
OneStopNoEnter2cor	79,79	73,39	1,81	21,19	18	33,33	66,67	0,00
OneStopNoEnter2cor	82,07	72,32	1,00	23,08	18	33,33	66,67	0,00
OneStopNoEnter2cor	75,92	72,44	2,18	21,78	18	33,33	66,67	0,00
OneStopNoEnter2cor	77,19	72,23	2,11	22,06	18	33,33	66,67	0,00
ShopAssistant1cor	65,00	66,09	8,29	24,82	12	42,86	57,14	0,00
ShopAssistant1cor	66,07	65,78	6,69	26,98	8	42,86	57,14	0,00
ShopAssistant1cor	70,80	69,38	5,99	23,03	8	42,86	57,14	0,00

ShopAssistant1cor	69,99	65,64	6,24	26,51	8	42,86	57,14	0,00
ShopAssistant1cor	72,05	62,11	7,08	29,21	8	42,86	57,14	0,00
ShopAssistant1cor	67,37	64,74	6,08	27,58	8	42,86	57,14	0,00
ShopAssistant1cor	71,14	61,77	7,04	29,59	8	42,86	57,14	0,00
ShopAssistant1cor	72,55	68,91	7,05	22,45	8	42,86	57,14	0,00
ShopAssistant1cor	67,94	68,77	5,87	23,76	8	42,86	57,14	0,00
ShopAssistant1cor	66,11	63,47	7,37	27,56	8	42,86	57,14	0,00
ShopAssistant2cor	64,66	42,18	4,73	52,26	47	16,67	72,22	11,11
ShopAssistant2cor	51,23	35,11	10,92	52,40	95	11,11	77,78	11,11
ShopAssistant2cor	53,74	24,36	11,40	56,24	40	16,67	72,22	11,11
ShopAssistant2cor	53,84	27,25	10,36	54,39	40	16,67	72,22	11,11
ShopAssistant2cor	51,66	29,80	11,61	50,59	40	16,67	72,22	11,11
ShopAssistant2cor	50,77	24,99	11,67	55,35	40	16,67	72,22	11,11
ShopAssistant2cor	51,51	30,49	9,20	52,32	40	16,67	72,22	11,11
ShopAssistant2cor	52,96	31,30	9,42	51,28	40	16,67	72,22	11,11
ShopAssistant2cor	52,35	28,89	9,77	53,35	40	16,67	72,22	11,11
ShopAssistant2cor	51,56	27,25	10,70	54,05	40	16,67	72,22	11,11
ThreePastShop1cor	64,32	26,25	9,16	63,47	40	0,00	61,54	38,46
ThreePastShop1cor	63,22	25,96	7,92	64,26	32	0,00	61,54	38,46
ThreePastShop1cor	62,49	22,73	7,19	63,69	32	0,00	61,54	38,46
ThreePastShop1cor	62,68	19,59	9,39	64,62	32	0,00	61,54	38,46
ThreePastShop1cor	62,22	18,87	6,90	67,83	32	0,00	61,54	38,46
ThreePastShop1cor	62,46	20,74	6,80	66,06	32	0,00	61,54	38,46
ThreePastShop1cor	65,43	20,73	9,65	63,21	32	0,00	61,54	38,46
ThreePastShop1cor	63,57	20,20	10,04	63,36	32	0,00	61,54	38,46
ThreePastShop1cor	61,81	23,29	7,59	62,71	32	0,00	61,54	38,46
ThreePastShop1cor	64,56	24,22	5,93	63,46	32	0,00	61,54	38,46
TwoEnterShop1cor	62,34	24,94	9,97	64,11	26	25,00	41,67	33,33
TwoEnterShop1cor	59,19	23,52	9,39	65,01	53	16,67	50,00	33,33
TwoEnterShop1cor	61,96	24,62	8,08	63,10	21	16,67	50,00	33,33
TwoEnterShop1cor	58,70	19,66	10,43	65,72	21	16,67	50,00	33,33
TwoEnterShop1cor	57,82	21,49	10,33	63,98	21	16,67	50,00	33,33
TwoEnterShop1cor	62,32	20,55	10,43	64,82	21	16,67	50,00	33,33
TwoEnterShop1cor	58,69	21,76	10,62	63,42	21	16,67	50,00	33,33
TwoEnterShop1cor	59,24	22,74	9,55	63,50	21	16,67	50,00	33,33
TwoEnterShop1cor	59,80	19,95	10,72	65,12	21	16,67	50,00	33,33
TwoEnterShop1cor	60,46	23,48	8,99	63,33	21	16,67	50,00	33,33
TwoEnterShop3cor	46,97	11,07	5,18	82,73	11	12,50	37,50	50,00
TwoEnterShop3cor	52,38	10,67	5,07	82,95	14	0,00	50,00	50,00
TwoEnterShop3cor	52,60	12,38	3,83	81,99	9	0,00	50,00	50,00
TwoEnterShop3cor	57,19	14,50	2,91	80,79	9	0,00	50,00	50,00
TwoEnterShop3cor	57,07	12,92	2,78	82,51	9	0,00	50,00	50,00
TwoEnterShop3cor	56,28	9,29	6,74	82,17	9	0,00	50,00	50,00
TwoEnterShop3cor	52,81	12,14	5,94	80,12	9	0,00	50,00	50,00
TwoEnterShop3cor	56,32	13,10	3,29	81,82	9	0,00	50,00	50,00
TwoEnterShop3cor	53,96	12,82	4,51	80,86	9	0,00	50,00	50,00
TwoEnterShop3cor	57,98	10,36	6,88	80,96	9	0,00	50,00	50,00
TwoLeaveShop2cor	66,41	51,91	13,31	34,78	15	50,00	50,00	0,00
TwoLeaveShop2cor	67,04	51,49	11,43	37,08	15	50,00	50,00	0,00
TwoLeaveShop2cor	71,68	48,29	11,43	38,31	15	50,00	50,00	0,00

TwoLeaveShop2cor	65,71	46,49	12,00	38,51	15	50,00	50,00	0,00
TwoLeaveShop2cor	71,57	50,46	8,55	38,00	15	50,00	50,00	0,00
TwoLeaveShop2cor	68,87	55,45	7,35	34,20	15	50,00	50,00	0,00
TwoLeaveShop2cor	67,63	57,28	6,81	32,90	15	50,00	50,00	0,00
TwoLeaveShop2cor	68,60	55,36	6,65	34,99	15	50,00	50,00	0,00
TwoLeaveShop2cor	72,53	53,19	7,39	36,43	15	50,00	50,00	0,00
TwoLeaveShop2cor	70,71	50,46	12,06	34,48	15	50,00	50,00	0,00
WalkByShop1cor	57,38	21,21	3,30	74,59	25	11,76	47,06	41,18
WalkByShop1cor	49,11	14,91	11,34	74,46	25	11,76	47,06	41,18
WalkByShop1cor	52,02	7,97	12,83	74,20	25	11,76	47,06	41,18
WalkByShop1cor	51,59	11,12	13,38	70,50	25	11,76	47,06	41,18
WalkByShop1cor	51,34	7,95	13,58	73,47	25	11,76	47,06	41,18
WalkByShop1cor	48,97	10,34	12,18	72,47	25	11,76	47,06	41,18
WalkByShop1cor	53,91	6,00	14,48	74,52	25	11,76	47,06	41,18
WalkByShop1cor	54,75	5,27	13,30	76,43	25	11,76	47,06	41,18
WalkByShop1cor	50,26	9,56	14,00	71,44	25	11,76	47,06	41,18
WalkByShop1cor	52,18	11,21	10,86	72,94	25	11,76	47,06	41,18

2. RAMUK-DL

SEQUÊNCIA	MOTP (%)	MOTA (%)	FP (%)	FN (%)	EC	MT (%)	PT (%)	ML (%)
PETS-2009 S2L1	65,36	89,35	2,18	7,89	29	84,21	15,79	0,00
PETS-2009 S2L1	72,81	89,54	2,58	7,30	29	84,21	15,79	0,00
PETS-2009 S2L1	66,65	91,24	2,02	6,16	29	84,21	15,79	0,00
PETS-2009 S2L1	73,66	89,73	2,43	7,25	29	84,21	15,79	0,00
PETS-2009 S2L1	64,87	89,44	2,79	7,19	29	84,21	15,79	0,00
PETS-2009 S2L1	74,14	89,76	2,62	7,04	29	84,21	15,79	0,00
PETS-2009 S2L1	71,92	91,55	1,67	6,20	29	84,21	15,79	0,00
PETS-2009 S2L1	64,11	91,02	0,48	7,92	29	84,21	15,79	0,00
PETS-2009 S2L1	66,48	90,58	0,31	8,53	29	84,21	15,79	0,00
PETS-2009 S2L1	73,43	92,61	1,26	5,55	29	84,21	15,79	0,00
EnterExitCrossingPaths1cor	53,46	29,26	12,77	57,37	3	40,00	20,00	40,00
EnterExitCrossingPaths1cor	52,65	34,51	13,51	51,38	3	40,00	20,00	40,00
EnterExitCrossingPaths1cor	61,80	38,93	8,04	52,44	3	40,00	20,00	40,00
EnterExitCrossingPaths1cor	62,35	33,31	9,99	56,11	3	40,00	20,00	40,00
EnterExitCrossingPaths1cor	55,50	28,93	13,95	56,52	3	40,00	20,00	40,00
EnterExitCrossingPaths1cor	62,05	38,44	11,38	49,58	3	40,00	20,00	40,00
EnterExitCrossingPaths1cor	54,34	37,95	5,81	55,64	3	40,00	20,00	40,00
EnterExitCrossingPaths1cor	62,42	33,24	8,58	57,58	3	40,00	20,00	40,00
EnterExitCrossingPaths1cor	56,44	40,91	5,28	53,21	3	40,00	20,00	40,00
EnterExitCrossingPaths1cor	56,43	32,53	9,52	57,35	3	40,00	20,00	40,00
EnterExitCrossingPaths2cor	73,86	77,90	8,77	13,13	1	100,00	0,00	0,00
EnterExitCrossingPaths2cor	71,98	79,75	6,57	13,47	1	100,00	0,00	0,00
EnterExitCrossingPaths2cor	77,95	77,03	10,08	12,69	1	100,00	0,00	0,00
EnterExitCrossingPaths2cor	80,07	76,29	9,28	14,23	1	100,00	0,00	0,00
EnterExitCrossingPaths2cor	72,81	78,79	7,81	13,20	1	100,00	0,00	0,00
EnterExitCrossingPaths2cor	78,05	77,85	9,24	12,71	1	100,00	0,00	0,00
EnterExitCrossingPaths2cor	81,08	81,83	3,68	14,29	1	100,00	0,00	0,00
EnterExitCrossingPaths2cor	80,90	79,52	9,21	11,07	1	100,00	0,00	0,00
EnterExitCrossingPaths2cor	78,55	80,32	9,21	10,27	1	100,00	0,00	0,00
EnterExitCrossingPaths2cor	74,62	82,41	3,56	13,83	1	100,00	0,00	0,00
OneLeaveShop1cor	44,96	36,13	2,32	60,95	3	0,00	60,00	40,00
OneLeaveShop1cor	46,04	37,40	1,74	60,26	3	0,00	60,00	40,00
OneLeaveShop1cor	52,95	41,64	1,17	56,59	3	0,00	60,00	40,00
OneLeaveShop1cor	48,63	41,88	3,34	54,18	3	0,00	60,00	40,00
OneLeaveShop1cor	47,43	41,56	3,56	54,29	3	0,00	60,00	40,00
OneLeaveShop1cor	45,47	34,27	3,21	61,92	3	0,00	60,00	40,00
OneLeaveShop1cor	45,70	37,29	3,17	58,94	3	0,00	60,00	40,00
OneLeaveShop1cor	48,19	34,69	2,02	62,68	3	0,00	60,00	40,00
OneLeaveShop1cor	43,93	45,15	1,16	53,08	3	0,00	60,00	40,00
OneLeaveShop1cor	45,01	39,18	0,40	59,82	3	0,00	60,00	40,00
OneLeaveShopReenter2cor	59,50	19,20	11,98	67,82	5	16,67	16,67	66,67
OneLeaveShopReenter2cor	64,58	20,28	10,89	67,83	5	16,67	16,67	66,67
OneLeaveShopReenter2cor	59,69	25,32	13,23	60,46	5	16,67	16,67	66,67
OneLeaveShopReenter2cor	63,78	20,59	12,19	66,22	5	16,67	16,67	66,67
OneLeaveShopReenter2cor	66,71	17,28	13,58	68,13	5	16,67	16,67	66,67
OneLeaveShopReenter2cor	66,06	23,78	13,32	61,90	5	16,67	16,67	66,67

OneLeaveShopReenter2cor	65,84	27,58	11,66	59,77	5	16,67	16,67	66,67
OneLeaveShopReenter2cor	65,41	20,51	12,32	66,17	5	16,67	16,67	66,67
OneLeaveShopReenter2cor	64,71	22,13	10,98	65,89	5	16,67	16,67	66,67
OneLeaveShopReenter2cor	60,29	19,31	11,53	68,16	5	16,67	16,67	66,67
OneShopOneWait1cor	55,97	45,23	0,96	50,81	15	14,29	57,14	28,57
OneShopOneWait1cor	58,19	40,34	3,69	52,97	15	14,29	57,14	28,57
OneShopOneWait1cor	57,06	43,73	1,37	51,90	15	14,29	57,14	28,57
OneShopOneWait1cor	56,21	45,51	2,96	48,52	15	14,29	57,14	28,57
OneShopOneWait1cor	61,64	46,98	3,58	46,43	15	14,29	57,14	28,57
OneShopOneWait1cor	56,57	48,31	-0,03	48,72	15	14,29	57,14	28,57
OneShopOneWait1cor	61,36	45,17	4,22	47,61	15	14,29	57,14	28,57
OneShopOneWait1cor	61,47	48,92	0,51	47,57	15	14,29	57,14	28,57
OneShopOneWait1cor	56,92	47,31	2,76	46,93	15	14,29	57,14	28,57
OneShopOneWait1cor	62,33	46,20	4,72	46,08	15	14,29	57,14	28,57
OneShopOneWait2cor	47,05	13,05	7,99	57,76	106	10,00	70,00	20,00
OneShopOneWait2cor	53,85	11,86	8,65	58,29	106	10,00	70,00	20,00
OneShopOneWait2cor	53,70	22,71	5,23	50,86	106	10,00	70,00	20,00
OneShopOneWait2cor	51,02	14,20	8,69	55,92	106	10,00	70,00	20,00
OneShopOneWait2cor	53,25	14,66	6,66	57,48	106	10,00	70,00	20,00
OneShopOneWait2cor	45,83	19,29	7,52	51,99	106	10,00	70,00	20,00
OneShopOneWait2cor	55,18	11,91	8,90	57,99	106	10,00	70,00	20,00
OneShopOneWait2cor	48,56	14,00	6,70	58,10	106	10,00	70,00	20,00
OneShopOneWait2cor	51,94	14,58	9,93	54,29	106	10,00	70,00	20,00
OneShopOneWait2cor	53,60	22,95	5,53	50,31	106	10,00	70,00	20,00
OneStopEnter1cor	76,32	54,14	2,83	41,64	7	33,33	50,00	16,67
OneStopEnter1cor	76,47	54,14	3,81	40,65	7	33,33	50,00	16,67
OneStopEnter1cor	84,05	53,89	1,51	43,20	7	33,33	50,00	16,67
OneStopEnter1cor	75,90	53,70	1,70	43,19	7	33,33	50,00	16,67
OneStopEnter1cor	78,83	56,17	1,39	41,03	7	33,33	50,00	16,67
OneStopEnter1cor	79,20	59,42	1,61	37,57	7	33,33	50,00	16,67
OneStopEnter1cor	80,77	60,53	1,55	36,52	7	33,33	50,00	16,67
OneStopEnter1cor	75,59	54,84	3,47	40,30	7	33,33	50,00	16,67
OneStopEnter1cor	81,16	58,38	3,60	36,62	7	33,33	50,00	16,67
OneStopEnter1cor	76,57	59,94	4,39	34,27	7	33,33	50,00	16,67
OneStopEnter2cor	78,96	62,50	16,44	20,86	1	100,00	0,00	0,00
OneStopEnter2cor	82,20	75,77	7,15	16,88	1	100,00	0,00	0,00
OneStopEnter2cor	79,42	74,09	9,42	16,29	1	100,00	0,00	0,00
OneStopEnter2cor	79,44	69,10	13,68	17,02	1	100,00	0,00	0,00
OneStopEnter2cor	78,07	71,73	15,02	13,05	1	100,00	0,00	0,00
OneStopEnter2cor	77,25	66,44	12,18	21,18	1	100,00	0,00	0,00
OneStopEnter2cor	78,43	65,40	16,82	17,58	1	100,00	0,00	0,00
OneStopEnter2cor	72,81	77,88	10,33	11,60	1	100,00	0,00	0,00
OneStopEnter2cor	79,22	68,45	15,45	15,90	1	100,00	0,00	0,00
OneStopEnter2cor	75,99	71,96	15,28	12,55	1	100,00	0,00	0,00
OneStopMoveEnter1cor	55,94	15,28	2,39	65,73	166	10,53	68,42	21,05
OneStopMoveEnter1cor	59,01	16,00	2,36	65,04	166	10,53	68,42	21,05
OneStopMoveEnter1cor	57,36	14,75	1,49	67,16	166	10,53	68,42	21,05
OneStopMoveEnter1cor	64,46	14,21	3,20	65,99	166	10,53	68,42	21,05
OneStopMoveEnter1cor	62,69	12,57	3,10	67,72	166	10,53	68,42	21,05
OneStopMoveEnter1cor	59,91	13,75	2,06	67,59	166	10,53	68,42	21,05

OneStopMoveEnter1cor	58,90	14,52	2,09	66,80	166	10,53	68,42	21,05
OneStopMoveEnter1cor	63,26	16,78	2,06	64,56	166	10,53	68,42	21,05
OneStopMoveEnter1cor	59,05	13,54	3,22	66,64	166	10,53	68,42	21,05
OneStopMoveEnter1cor	56,42	15,06	2,01	66,33	166	10,53	68,42	21,05
OneStopMoveEnter2cor	62,93	17,89	11,23	69,68	6	27,27	0,00	72,73
OneStopMoveEnter2cor	67,70	17,76	9,35	71,69	6	27,27	0,00	72,73
OneStopMoveEnter2cor	71,50	17,57	10,90	70,34	6	27,27	0,00	72,73
OneStopMoveEnter2cor	67,89	20,85	9,89	68,06	6	27,27	0,00	72,73
OneStopMoveEnter2cor	63,64	16,34	10,89	71,57	6	27,27	0,00	72,73
OneStopMoveEnter2cor	64,14	17,89	9,66	71,25	6	27,27	0,00	72,73
OneStopMoveEnter2cor	71,61	17,84	11,32	69,64	6	27,27	0,00	72,73
OneStopMoveEnter2cor	65,60	17,49	11,56	69,75	6	27,27	0,00	72,73
OneStopMoveEnter2cor	61,55	18,97	10,43	69,40	6	27,27	0,00	72,73
OneStopMoveEnter2cor	64,55	18,08	9,84	70,88	6	27,27	0,00	72,73
OneStopMoveNoEnter1cor	55,64	23,85	5,58	70,57	0	25,00	25,00	50,00
OneStopMoveNoEnter1cor	55,75	29,26	7,35	63,39	0	25,00	25,00	50,00
OneStopMoveNoEnter1cor	60,48	32,57	4,79	62,64	0	25,00	25,00	50,00
OneStopMoveNoEnter1cor	58,81	31,37	3,88	64,75	0	25,00	25,00	50,00
OneStopMoveNoEnter1cor	61,97	23,98	4,26	71,76	0	25,00	25,00	50,00
OneStopMoveNoEnter1cor	53,77	32,52	4,00	63,48	0	25,00	25,00	50,00
OneStopMoveNoEnter1cor	60,42	28,36	2,71	68,94	0	25,00	25,00	50,00
OneStopMoveNoEnter1cor	60,28	21,62	7,25	71,13	0	25,00	25,00	50,00
OneStopMoveNoEnter1cor	56,79	21,71	7,52	70,77	0	25,00	25,00	50,00
OneStopMoveNoEnter1cor	56,51	23,35	5,96	70,69	0	25,00	25,00	50,00
OneStopNoEnter1cor	78,11	56,15	1,87	41,57	2	66,67	0,00	33,33
OneStopNoEnter1cor	76,44	55,91	2,29	41,40	2	66,67	0,00	33,33
OneStopNoEnter1cor	80,50	59,67	1,28	38,66	2	66,67	0,00	33,33
OneStopNoEnter1cor	80,88	59,13	1,62	38,86	2	66,67	0,00	33,33
OneStopNoEnter1cor	76,21	59,04	1,31	39,26	2	66,67	0,00	33,33
OneStopNoEnter1cor	77,98	57,89	1,51	40,20	2	66,67	0,00	33,33
OneStopNoEnter1cor	86,64	56,49	2,08	41,03	2	66,67	0,00	33,33
OneStopNoEnter1cor	76,54	57,11	2,21	40,28	2	66,67	0,00	33,33
OneStopNoEnter1cor	77,59	53,29	2,40	43,91	2	66,67	0,00	33,33
OneStopNoEnter1cor	80,71	59,37	2,08	38,15	2	66,67	0,00	33,33
OneStopNoEnter2cor	75,85	71,82	2,04	22,54	18	33,33	66,67	0,00
OneStopNoEnter2cor	77,44	73,66	1,24	21,50	18	33,33	66,67	0,00
OneStopNoEnter2cor	74,32	74,72	1,09	20,59	18	33,33	66,67	0,00
OneStopNoEnter2cor	75,59	77,86	0,04	18,50	18	33,33	66,67	0,00
OneStopNoEnter2cor	75,08	72,46	0,39	23,56	18	33,33	66,67	0,00
OneStopNoEnter2cor	80,36	73,24	2,22	20,95	18	33,33	66,67	0,00
OneStopNoEnter2cor	78,84	70,01	2,25	24,14	18	33,33	66,67	0,00
OneStopNoEnter2cor	74,50	70,23	1,88	24,29	18	33,33	66,67	0,00
OneStopNoEnter2cor	78,67	73,60	0,76	22,04	18	33,33	66,67	0,00
OneStopNoEnter2cor	74,41	75,98	0,72	19,71	18	33,33	66,67	0,00
ShopAssistant1cor	66,07	64,73	6,69	26,98	8	42,86	57,14	0,00
ShopAssistant1cor	67,65	70,11	4,93	23,36	8	42,86	57,14	0,00
ShopAssistant1cor	67,26	65,05	7,26	26,09	8	42,86	57,14	0,00
ShopAssistant1cor	69,78	70,10	6,24	22,05	8	42,86	57,14	0,00
ShopAssistant1cor	66,32	62,15	7,55	28,71	8	42,86	57,14	0,00
ShopAssistant1cor	65,47	68,94	4,99	24,48	8	42,86	57,14	0,00

ShopAssistant1cor	64,37	62,62	7,59	28,19	8	42,86	57,14	0,00
ShopAssistant1cor	70,47	71,59	7,55	19,25	8	42,86	57,14	0,00
ShopAssistant1cor	70,11	63,95	6,24	28,21	8	42,86	57,14	0,00
ShopAssistant1cor	66,54	68,33	4,70	25,37	8	42,86	57,14	0,00
ShopAssistant2cor	65,87	40,02	4,72	47,26	40	16,67	72,22	11,11
ShopAssistant2cor	67,56	40,22	2,99	48,78	40	16,67	72,22	11,11
ShopAssistant2cor	67,89	37,90	4,47	49,63	40	16,67	72,22	11,11
ShopAssistant2cor	68,27	40,69	3,96	47,34	40	16,67	72,22	11,11
ShopAssistant2cor	69,86	37,42	5,27	49,31	40	16,67	72,22	11,11
ShopAssistant2cor	65,33	37,28	3,05	51,67	40	16,67	72,22	11,11
ShopAssistant2cor	69,77	38,44	4,13	49,43	40	16,67	72,22	11,11
ShopAssistant2cor	70,85	34,36	5,55	52,09	40	16,67	72,22	11,11
ShopAssistant2cor	65,82	34,65	5,32	52,03	40	16,67	72,22	11,11
ShopAssistant2cor	68,00	42,06	3,87	46,07	40	16,67	72,22	11,11
ThreePastShop1cor	66,43	23,31	9,16	61,13	32	0,00	61,54	38,46
ThreePastShop1cor	67,23	20,49	8,55	64,56	32	0,00	61,54	38,46
ThreePastShop1cor	68,86	20,58	8,79	64,23	32	0,00	61,54	38,46
ThreePastShop1cor	70,88	21,41	7,79	64,40	32	0,00	61,54	38,46
ThreePastShop1cor	66,41	14,65	11,98	66,97	32	0,00	61,54	38,46
ThreePastShop1cor	68,11	20,37	7,84	65,39	32	0,00	61,54	38,46
ThreePastShop1cor	64,83	18,08	12,06	63,46	32	0,00	61,54	38,46
ThreePastShop1cor	71,18	22,19	10,19	61,22	32	0,00	61,54	38,46
ThreePastShop1cor	71,81	22,55	9,37	61,68	32	0,00	61,54	38,46
ThreePastShop1cor	66,93	17,77	9,72	66,11	32	0,00	61,54	38,46
TwoEnterShop1cor	61,82	22,75	9,15	63,90	21	25,00	41,67	33,33
TwoEnterShop1cor	59,88	23,87	9,73	62,21	21	25,00	41,67	33,33
TwoEnterShop1cor	64,38	23,58	7,16	65,06	21	25,00	41,67	33,33
TwoEnterShop1cor	65,75	23,64	7,20	64,96	21	25,00	41,67	33,33
TwoEnterShop1cor	63,28	23,95	8,58	63,27	21	25,00	41,67	33,33
TwoEnterShop1cor	64,36	21,89	9,20	64,71	21	25,00	41,67	33,33
TwoEnterShop1cor	61,74	22,24	11,09	62,47	21	25,00	41,67	33,33
TwoEnterShop1cor	61,39	22,11	10,30	63,38	21	25,00	41,67	33,33
TwoEnterShop1cor	60,00	22,61	8,35	64,84	21	25,00	41,67	33,33
TwoEnterShop1cor	64,85	21,92	9,73	64,16	21	25,00	41,67	33,33
TwoEnterShop3cor	52,38	12,38	4,88	80,94	9	0,00	50,00	50,00
TwoEnterShop3cor	54,99	9,84	6,68	81,68	9	0,00	50,00	50,00
TwoEnterShop3cor	56,41	14,05	5,21	78,94	9	0,00	50,00	50,00
TwoEnterShop3cor	52,66	11,13	6,42	80,65	9	0,00	50,00	50,00
TwoEnterShop3cor	57,20	11,61	3,85	82,74	9	0,00	50,00	50,00
TwoEnterShop3cor	54,58	12,11	3,22	82,87	9	0,00	50,00	50,00
TwoEnterShop3cor	52,66	11,77	6,20	80,24	9	0,00	50,00	50,00
TwoEnterShop3cor	57,25	12,62	4,70	80,88	9	0,00	50,00	50,00
TwoEnterShop3cor	51,24	13,41	6,76	78,02	9	0,00	50,00	50,00
TwoEnterShop3cor	53,74	13,98	3,06	81,16	9	0,00	50,00	50,00
TwoLeaveShop2cor	67,04	48,29	11,43	37,08	16	50,00	50,00	0,00
TwoLeaveShop2cor	67,39	47,90	13,12	35,77	16	50,00	50,00	0,00
TwoLeaveShop2cor	72,57	47,37	11,27	38,17	16	50,00	50,00	0,00
TwoLeaveShop2cor	68,85	52,67	10,12	34,01	16	50,00	50,00	0,00
TwoLeaveShop2cor	69,97	46,48	12,72	37,60	16	50,00	50,00	0,00
TwoLeaveShop2cor	69,61	53,34	7,59	35,88	16	50,00	50,00	0,00

TwoLeaveShop2cor	65,91	55,12	7,54	34,14	16	50,00	50,00	0,00
TwoLeaveShop2cor	70,59	51,80	9,80	35,20	16	50,00	50,00	0,00
TwoLeaveShop2cor	65,39	46,15	13,13	37,52	16	50,00	50,00	0,00
TwoLeaveShop2cor	66,99	50,98	12,86	32,96	16	50,00	50,00	0,00
WalkByShop1cor	49,11	9,25	9,50	74,46	34	0,00	41,18	58,82
WalkByShop1cor	53,08	10,84	11,38	70,98	34	0,00	41,18	58,82
WalkByShop1cor	47,18	6,55	11,39	75,26	34	0,00	41,18	58,82
WalkByShop1cor	48,09	9,54	10,05	73,60	34	0,00	41,18	58,82
WalkByShop1cor	47,59	13,04	8,75	71,40	34	0,00	41,18	58,82
WalkByShop1cor	52,40	7,56	9,43	76,22	34	0,00	41,18	58,82
WalkByShop1cor	52,80	10,23	9,75	73,21	34	0,00	41,18	58,82
WalkByShop1cor	52,67	7,54	10,37	75,28	34	0,00	41,18	58,82
WalkByShop1cor	52,84	8,81	9,03	75,36	34	0,00	41,18	58,82
WalkByShop1cor	47,27	8,82	8,78	75,60	34	0,00	41,18	58,82
