



Rastreamento de Objetos por Similaridade de Características: Uma Aplicação para Compressão e Indexação de Vídeos

Trabalho de Conclusão de Curso

Engenharia da Computação

Aluno: André Hermenegildo do Nascimento

Orientador: Prof. Bruno José Torres Fernandes



**Universidade de Pernambuco
Escola Politécnica de Pernambuco
Graduação em Engenharia de Computação**

André Hermenegildo do Nascimento

**Rastreamento de Objetos por
Similaridade de Características: Uma
Aplicação para Compressão e
Indexação de Vídeos**

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia de Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

Recife, Junho de 2011.

MONOGRAFIA DE FINAL DE CURSO

Avaliação Final (para o presidente da banca)*

No dia 15 de Junho de 2011, às 15:00 horas, reuniu-se para deliberar a defesa da monografia de conclusão de curso do discente ANDRE HERMENEGILDO DO NASCIMENTO, orientado pelo professor Bruno José Torres Fernandes, sob título Rastreamento de objetos por similaridade de características: uma aplicação para compressão e indexação de vídeos, a banca composta pelos professores:

Carmelo Bastos Filho

Bruno José Torres Fernandes

George Darminton

Após a apresentação da monografia e discussão entre os membros da Banca, a mesma foi considerada:

Aprovada Aprovada com Restrições* Reprovada

e foi-lhe atribuída nota: 9,5 (NOVE E MEIO)

*(Obrigatório o preenchimento do campo abaixo com comentários para o autor)

O discente terá 7 dias para entrega da versão final da monografia a contar da data deste documento.


CARMELO BASTOS FILHO


BRUNO JOSÉ TORRES FERNANDES


GEORGE DARMINTON

* Este documento e sua cópia deverão ser encadernados juntamente com às monografias em versão final.

Dedicatória

Dedico aos meus pais, Alúcio Hermenegildo e Maria José, e às mulheres da minha vida, minha amada esposa Tarciana Chalegre e nossa filha Beatriz Chalegre. Todos meus esforços só fazem sentido porque tenho vocês ao meu lado.

Agradecimentos

Agradeço a Deus por todas as graças que venho alcançando e por além de ter me dado força e inspiração para a conclusão deste trabalho, ter colocado pessoas tão especiais em minha vida.

Agradeço a meus pais, meus irmãos, Alexandre Hermenegildo e Alexandra Hermenegildo, por todo apoio, incentivo e por acreditarem no meu potencial. O meu muito obrigado por sermos essa família tão unida.

Um agradecimento muito especial a minha esposa, amiga, corretora extraoficial, fonte de motivação e apoio, por todo companheirismo durante toda esta jornada acadêmica e compreensão dos vários momentos que não estive presente, principalmente durante a realização deste trabalho.

Agradeço a minha filhinha linda, por me alegrar nos momentos mais difíceis, e me acalmar quando mais precisei. Obrigado Bia, por ser este instrumento de Deus em minha vida.

Agradeço a todos os amigos conquistados durante esses cinco anos de muitas dificuldades, provas e projetos. Em especial a Anderson Murilo, Thiago Martins, Gabriel Baracuh, Carlos Eduardo, Bruno Carlos, Francisco Marinho, Marcos Antônio, Leandro Honorato, Jefferson Amorim, Péricles Miranda, Rodrigo Mendes, Diogo Gregório e Adriano Marçal, pela amizade que vai muito além das fronteiras de uma sala de aula, além de todos da turma ec2006.2.

Agradeço também a Hítalo Oliveira, por ter sido um grande companheiro durante toda essa jornada, principalmente nos últimos anos, foi uma pessoa com quem sempre pude contar e com quem aprendi e cresci muito.

Agradeço a todos os professores que contribuíram com a minha formação acadêmica. Dentre eles os professores, Marcio Cornélio, Sérgio Campello, Sérgio Murilo, Wellington Pinheiro, Byron Leite e Mêuser Jorge, além de agradecer a todos que lutam para fazer do e-Comp um ótimo curso. Agradeço aqui também a meu orientador Bruno Fernandes, por total disponibilidade, dedicação, correções e contribuições durante a realização desta monografia.

Por fim, gostaria de agradecer a todos as pessoas que direta ou indiretamente contribuíram para o meu sucesso nesta realização.

Resumo

Rastreamento de objetos é apresentado como tema bastante promissor dentro da visão computacional. Possui considerável importância prática em diversas aplicações. Como a detecção automática de incidentes, o acompanhamento das principais redes rodoviárias, shopping centers e lojas, controle inteligente de velocidade para veículos em trânsito, entre outras. Este trabalho propõe uma abordagem para o rastreamento de objetos por similaridade de características, visando o desenvolvimento de um método de compressão e indexação de vídeos de maneira a melhor gerenciar a grande quantidade de vídeos gerados por sistemas de monitoramento e rastreamento de objetos. Desta forma, foi realizado um levantamento das técnicas necessárias para o desenvolvimento de um protótipo de sistema de rastreamento, que foi utilizado para a criação de um novo método de compressão e indexação de vídeos. Resultados promissores foram alcançados, o que mostrou aplicabilidade do método em inúmeros sistemas.

Abstract

Object tracking is presented as a promising topic in computer vision. It has been considered of practical importance in many applications. Like the automatic incident detection, monitoring of major roads, shopping centers and stores, intelligent control of speed for transit vehicles, among others. This study proposes an approach for tracking objects based on the similarity of their features, aiming the development of a method for video compression and indexing in order to improve the management of a large amount of video generated by monitoring systems and object tracking. Thus, a survey about the techniques necessary for the development of a prototype tracking system, which was used to create a novel method for compression and indexing of videos. Very promising results were obtained which demonstrated the applicability of the method in many systems.

Sumário

Capítulo 1	Introdução	1
1.1	Objetivos	2
	Objetivos Gerais:	2
	Objetivos Específicos:	3
1.2	Estrutura da Monografia	3
Capítulo 2	Segmentação de Movimento	4
2.1	Representação de Objetos	4
2.2	Segmentação de Objetos	6
2.2.1	Subtração de Imagens de Fundos	7
2.2.2	Diferença Temporal	11
2.2.3	Modelos estatísticos	11
2.3	Pós-Segmentação	14
2.3.1	Identificação de Regiões	15
2.3.2	Eliminação de Ruídos	15
Capítulo 3	Rastreamento de Objetos	17
3.1	Rastreamento de objetos por similaridades de características	17
3.2	Correspondência entre Objetos	21
3.3	Estados dos Objetos	24
3.4	Oclusões de Objetos (<i>Occlusion</i>)	26
3.5	Divisão de Objetos (<i>Splitting</i>)	26
Capítulo 4	Compressão e Indexação de Vídeo Baseado em Rastreamento de Objetos (CIVRO)	28
4.1	Visão Geral	28
4.2	Compressão	30
4.2.1	Codificador	32

4.2.2	Decodificador	36
Capítulo 5	Arranjo Experimental	38
5.1	Bases de imagens	38
5.2	Arquitetura do sistema	39
5.2.1	Captura de Imagens	40
5.2.2	Segmentação de Movimento	42
5.2.3	Rastreamento de objetos	42
5.2.4	Compressão CIVRO	45
5.2.5	Exibição	47
5.3	Análises dos Resultados	49
5.3.1	Segmentação de movimento	49
5.3.2	Rastreamento de Objetos	51
5.3.3	Compressão CIVRO	52
Capítulo 6	Conclusão e Trabalhos Futuros	55
6.1	Trabalhos Futuros	56
	Bibliografia	57
	Apêndice A Código Fonte	60

Índice de Figuras

Figura 2 - 1. Representação de Objetos. a) Centróide, B) Multipontos, c) Forma retangular, d) Forma elíptica, e) Forma articuladas, f) Esqueleto, g) Pontos de contorno, h) Fronteira de contorno, i) Silhueta (Figura extraída de [1]).	6
Figura 2 - 2. Exemplo de segmentação de objetos. a) <i>Background model</i> , B) Quadro original, c) Resultado do processo de segmentação de objetos.	7
Figura 2 - 3. Ruídos na segmentação causados por alteração na iluminação. (a) Modelo de fundo do ambiente, (b) Quadro de entrada para segmentação, (c) Quadro resultante da segmentação, (d) Identificação de regiões - retângulos mínimos, (e) Identificação de regiões - retângulos mínimos e centróide.	14
Figura 2 - 4. Exemplo de identificação de regiões utilizando a representação de retângulos mínimos e centróide.	15
Figura 2 - 5. Exemplo de eliminação de ruídos. (a) segmentação inicial, (b) fechamento de regiões, (c) identificação de regiões conectadas (em vermelho as regiões que serão descartadas), (d) Resultado da eliminação de regiões pequenas, (e) Identificação de regiões de interesses, (f) resultado final do processo de eliminação de ruídos.	16
Figura 3 - 1. Fases da etapa de casamento de características [3].	20
Figura 3 - 2. Correspondência Um-para-Um.	22
Figura 3 - 3. Correspondência Um-para-Muitos.	22
Figura 3 - 4. Correspondência Um-para-Nenhum.	23
Figura 3 - 5. Correspondência Nenhum-para-Um.	23
Figura 3 - 6. Correspondência Muitos-para-Um.	24
Figura 3 - 7. Diagrama de transição de estados assumidos por objetos em rastreamento [23].	24
Figura 4 - 1. Modelo do sistema proposto.	29
Figura 4 - 3. Diagrama de blocos do sistema de compressão.	31
Figura 4 - 4. Diagrama de blocos do codificador.	32

Figura 4 - 5. Modelo conceitual da estrutura de armazenamento.	33
Figura 4 - 6. Modelo lógico da estrutura de armazenamento.	34
Figura 4 - 7. Cenário com fundo constante.	35
Figura 4 - 8. Cenário com fundo constante e um mesmo objeto por vários quadros.	36
Figura 4 - 9. Diagrama de blocos do decodificador.	37
Figura 4 - 10. Exemplo de decodificação de imagens.	37
Figura 5 - 1. Modelo geral do sistema.	39
Figura 5 - 2. Exemplos imagens capturas. (a) PETSDATABASE2000, (b) PETSDATABASE2001, (c) PETSDATABASE2006.	40
Figura 5 - 3. Diagrama de atividades – Captura de Imagens.	41
Figura 5 - 4. Saída do módulo de Segmentação de Movimento. (a) PETSDATABASE2000, (b) PETSDATABASE2001, (c) PETSDATABASE2006.	42
Figura 5 - 5. Diagrama de atividades – Segmentação de Movimento.	43
Figura 5 - 6. Diagrama de atividades – Rastreamento de Objetos.	44
Figura 5 - 7. Diagrama de atividades – Compressão (Codificação).	45
Figura 5 - 8. Imagens codificadas. (a) PETSDATABASE2000, (b) PETSDATABASE2001, (c) PETSDATABASE2006.	46
Figura 5 - 9. Diagrama de atividades – Compressão (Decodificação).	46
Figura 5 - 10. Imagem de fundo de referência e os objetos codificados para o quadro. (a) PETSDATABASE2000, (b) PETSDATABASE2001, (c) PETSDATABASE2006.	47
Figura 5 - 11. Diagrama de atividades – Exibição.	48
Figura 5 - 12. Saída do módulo de Exibição.	48
Figura 5 - 13. Saída do módulo de Exibição. (a) PETSDATABASE2000, (b) PETSDATABASE2001, (c) PETSDATABASE2006.	49
Figura 5 - 14. Comparativo das técnicas de segmentação. (a) Segmentação usando Subtração de Fundo, (b) Segmentação por Diferença Temporal e (c) Segmentação por Modelo de Mistura de Gaussianas.	51

Índice de Tabelas

Tabela 3 - 1. Características utilizadas por Amer [8].	18
Tabela 5 - 1. Resumo das bases utilizadas.	39
Tabela 5 - 2. Compressão CIVRO.....	54

Tabela de Símbolos e Siglas

- CFTV - Circuito Fechado Televisão
- CIVRO - Compressão e Indexação de Vídeo Baseado em Rastreamento de Objetos
- JPEG - *Joint Photographic Experts Group*
- MPEG - *Moving Picture Experts Group*
- PETS - *Performance Evaluation of Tracking and Surveillance*

Capítulo 1

Introdução

O aumento da criminalidade trouxe a necessidade de aperfeiçoar o monitoramento de sistemas de vigilância. Some-se a isso a recente e rápida ampliação do número de câmeras CFTV (Circuito Fechado Televisão) e a existência de uma crescente necessidade de análise automatizada de vídeos. Esses fatos têm acelerado a demanda por técnicas automáticas de processamento e análise de imagens [1].

Neste contexto, o rastreamento de objetos [2-6] apresenta-se como uma peça chave dentro da visão computacional para o processo de interpretação automática e geração de semântica ou tradução em linguagem natural dos acontecimentos dos quadros dos vídeos. Esta técnica é utilizada desde sistemas de radares automotivos [7] a sistemas de vigilância automatizados para rastreamento de pessoas e detecção de eventos suspeitos [8].

Em se tratando de vigilância, o rastreamento é um tema bastante promissor, sendo base de estudos e pesquisas internacionais [9]. Ele possui considerável importância prática em diversas aplicações, como: a detecção automática de incidentes, o acompanhamento das principais redes rodoviárias, shopping centers e lojas, controle inteligente de velocidade para veículos em trânsito, entre outros.

Essas aplicações destacam-se como eficazes provas processuais criminais, visto que são capazes de identificar os autores de crimes, em virtude do armazenamento de suas imagens em bancos de dados. Além disso, possui uma ação preventiva, evitando que determinados crimes aconteçam, favorecendo o flagrante [9].

Têm sido propostas várias abordagens para o rastreamento de objetos, dentre elas o rastreamento por similaridade de características [3], [4], que consiste em estabelecer relações temporais entre objetos alvos de quadros de vídeos consecutivos com base na similaridade de características obtidas dos objetos destes quadros.

Em geral, os sistemas de rastreamento de objetos necessitam armazenar uma grande quantidade de vídeos digitais, pois, o ambiente é monitorado 24 horas por dia e as imagens geradas devem ser arquivadas por um longo período de tempo, em virtude da necessidade de eventuais consultas. Entretanto, grande parte dos dados gerados por esses sistemas acaba ficando sem utilidade [10], visto que não apresentam uma organização que permita a recuperação eficiente, sendo armazenados apenas por critérios de segurança.

Sendo assim, necessita-se de estratégias que gerenciem eficientemente o grande fluxo de informações geradas por sistemas de rastreamento de objetos, de maneira a maximizar o espaço de armazenamento e permitir agilidade na recuperação de imagens de cenas específicas.

1.1 Objetivos

Neste trabalho, é desenvolvido um protótipo de rastreamento de objetos capaz de rastrear objetos alvos em quadros consecutivos de vídeos utilizando similaridades de características. A partir da aquisição de objetos em quadros de vídeos, por exemplo, é possível realizar uma correspondência temporal entre os objetos de forma a traçar as trajetórias por eles realizadas. Uma forma de também colaborar com investigações criminais quando aplicada no contexto de vigilância.

Além disso, é proposta uma estratégia de armazenamento e recuperação das imagens obtidas visando uma economia do espaço utilizado para armazenamento e a agilidade na recuperação das imagens de cenas específicas.

Objetivos Gerais:

- Desenvolver um protótipo de rastreamento de objetos por similaridades de características;
- Propor uma abordagem de compressão de vídeos que permita um melhor gerenciamento dos vídeos gerados por sistemas de rastreamento de objetos.

Objetivos Específicos:

- Apresentar revisão da literatura acerca dos métodos de rastreamento, sobretudo, as técnicas de rastreamento por similaridade de características;
- Difundir o conhecimento das técnicas de rastreamento de objetos de forma a facilitar o surgimento de novas aplicações.

1.2 Estrutura da Monografia

Este trabalho está dividido em seis capítulos. No primeiro capítulo foi dada uma introdução aos sistemas de rastreamento de objetos e sua aplicabilidade em alguns sistemas. Os demais capítulos são apresentados a seguir:

- **Capítulo 2 – Segmentação de Movimento:** este capítulo aborda o conteúdo teórico necessário para o processo de segmentação de movimento, segmentação de objetos, identificações de regiões em movimento e eliminação de ruídos;
- **Capítulo 3 – Rastreamento de Objetos:** são apresentadas as principais técnicas para rastreamento de objetos por similaridade de características, correspondência entre objetos, problemas causados por oclusões e divisão de objetos e técnicas para detecção e tratamento desses problemas;
- **Capítulo 4 – Compressão e Indexação de Vídeo Baseado em Rastreamento de Objetos (CIVRO):** é apresentada uma estratégia para compressão de imagens para sistemas de rastreamentos de objetos visando economia do espaço utilizado para armazenamento e agilidade na recuperação de imagens de cenas específicas;
- **Capítulo 5 – Arranjo Experimental:** apresenta a arquitetura do sistema desenvolvido, os resultados dos experimentos realizados e análise das técnicas utilizadas;
 - **Capítulo 6 – Conclusão e Trabalhos Futuros:** é apresentada uma conclusão geral do trabalho, dificuldades encontradas, sugestões de trabalhos futuros.

Capítulo 2

Segmentação de Movimento

As técnicas de rastreamento de objetos necessitam, em sua maioria, de uma etapa de pré-processamento que consiste em separar e classificar os objetos em movimento no ambiente observado.

Normalmente, a segmentação de movimento é representada por um módulo detector de movimento, para o qual são enviadas as imagens capturadas pelas câmeras instaladas no ambiente observado [8]. Esse módulo tem como função extrair objetos em movimento do resto da cena. Além disso, é possível realizar melhoramentos nas imagens, como: eliminação de ruídos, detecção de regiões conectadas e identificação dos objetos alvos em movimento.

Neste capítulo será abordado todo o conteúdo teórico necessário para o processo de segmentação de movimento. A seção 2.1 mostra como é feita a representação de objetos no contexto de rastreamento de objetos. Em seguida, a seção 2.2 apresenta como é realizada a segmentação de objetos em movimento em uma cena, e por fim, a seção 2.3 discursa sobre técnicas de pós-segmentação que visam melhorar os resultados das imagens segmentadas.

2.1 Representação de Objetos

No cenário de rastreamento, um objeto é definido como algo de interesse para outras análises. Por exemplo, os barcos no mar, os peixes dentro de um aquário, os veículos em uma estrada, aviões no ar ou pessoas andando em uma estrada representam um conjunto de objetos importantes em um determinado domínio [1]. Objetos podem ser representados de acordo com propriedades de diferentes naturezas, como contorno, cor, textura e forma.

A representação dos objetos tem impacto direto no desempenho do rastreamento no domínio de interesse, visto que uma representação ruim pode ocasionar a eliminação de informações de relevância para o rastreamento, podendo até inviabilizar o processo.

A seguir são apresentadas as formas de representação de objetos comumente usadas no rastreamento e alguns domínios onde são mais indicadas.

- **Pontos** - O objeto é representado por um ponto, centróide (**Figura 2 - 1 (a)**) [11], ou por um conjunto de pontos, multipontos (**Figura 2 - 1 (b)**) [12]. Em geral, é adequado para representar objetos que ocupam pequenas regiões na imagem;
- **Formas Geométricas** - O objeto é representado por retângulos, elipses (**Figura 2 - 1 (c), (d)**) [13], e outras formas geométricas primitivas. É mais recomendada para representar objetos rígidos (objetos que apresentam sempre o mesmo formato. ex.: carros, barcos e demais inanimados), porém também pode ser utilizada para representação de objetos não-rígidos (ex.: animais e objetos de forma variável);
- **Formas Articuladas** - O objeto é representado por partes que são unidas por meios de articulações. O corpo humano é um exemplo elementar deste tipo de representação, onde se pode destacar cabeça, tronco e membros, além de suas divisões intermediárias (**Figura 2 - 1 (e)**);
- **Esqueleto** - O objeto é representado por meio de um esqueleto (**Figura 2 - 1 (f)**) que é obtido através de um algoritmo de afinamento (também conhecido como algoritmo de esqueletização) [14]. É utilizado para representação de objetos articulados e rígidos;
- **Silhueta e Contorno** - O objeto é representado por meio de uma fronteira que define o seu limite (**Figura 2 - 1 (g) (h)**). A região interna do contorno é conhecida de silhueta (**Figura 2 - 1 (i)**). Este tipo de representação é adequado para objetos não-rígidos;

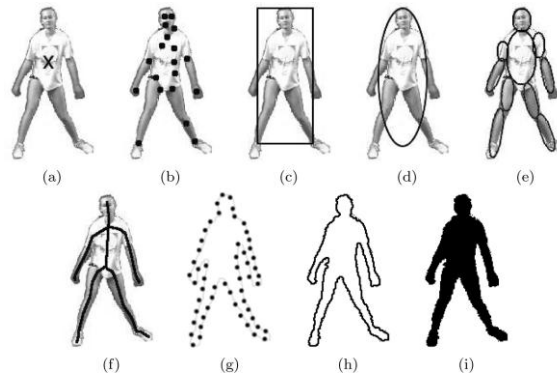


Figura 2 - 1. Representação de Objetos. a) Centróide, B) Multipontos, c) Forma retangular, d) Forma elíptica, e) Forma articuladas, f) Esqueleto, g) Pontos de contorno, h) Fronteira de contorno, i) Silhueta (Figura extraída de [1]).

2.2 Segmentação de Objetos

Cada método de rastreamento de objetos exige um mecanismo de segmentação, seja este realizado quadro a quadro ou quando o primeiro objeto aparecer. A segmentação consiste em separar os objetos em movimento de uma cena (*foreground*) da parte estática ou fundo da cena (*background*) [15].

Uma abordagem comum para segmentação de objetos consiste em manter um modelo de fundo da cena (*background model*), ou seja, um modelo do ambiente sem a presença de objetos em movimento [12], e realizar comparações com uma imagem de entrada, possibilitando classificar cada pixel do quadro de entrada como *foreground* ou *background*. Qualquer mudança significativa em uma região da imagem do modelo de fundo é tida como um movimento de objeto (*foreground*). Os pixels que constituem esta determinada região são marcados e, geralmente, é aplicado um algoritmo para agrupar as áreas de movimento mais próximas umas das outras, diminuindo assim problemas ocasionados por ruídos.

A segmentação de objetos é tida como uma tarefa difícil [8], pois diversos elementos podem interferir na confiabilidade e no desempenho das técnicas comumente usadas, como por exemplo: mudança na iluminação do ambiente, sombras, superfícies espelhadas, movimentos de vegetação, dentre outros. De maneira geral, não existe um algoritmo de segmentação de objetos robusto o suficiente para se adaptar a todos os fatores e alterações ambientais. Desta forma,

deve-se ter em mente o tipo de aplicação e o tipo de ambiente (interno ou externo) que será utilizado e escolher a técnica mais adequada.

De modo geral, quanto menor a influência de fatores externos ao ambiente maior será a robustez do algoritmo de segmentação. A **Figura 2 - 2** apresenta um exemplo de segmentação de objetos em um ambiente controlado. Nas subseções seguintes, serão apresentadas as principais abordagens de segmentação de objetos baseados na detecção de movimento e subtração de imagens de fundo [2].

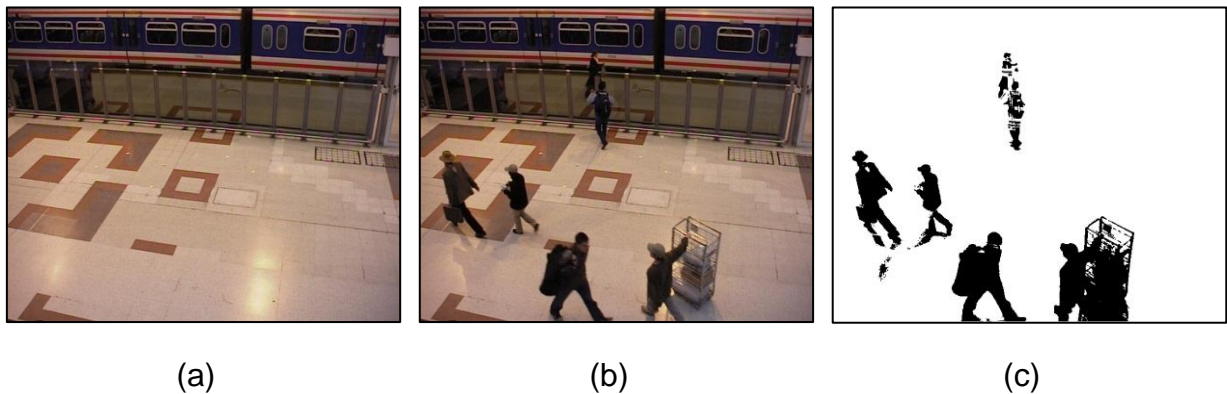


Figura 2 - 2. Exemplo de segmentação de objetos. a) *Background model*, B) Quadro original, c) Resultado do processo de segmentação de objetos.

2.2.1 Subtração de Imagens de Fundos

A subtração de fundo é uma técnica muito popular para a segmentação de objetos baseada em movimento [2], principalmente em situações com um fundo relativamente estático. Ela tenta detectar as regiões que se deslocam em uma imagem através da diferença entre a imagem atual e uma imagem de fundo de referência, se esta diferença for maior que um limiar, este pixel é considerado como pertencente ao alvo, caso contrário como pertencente ao fundo. É um método relativamente simples e de fácil aplicação, apresentando uma boa recuperação da forma do alvo. No entanto, é extremamente sensível a mudanças de cenas, devido à iluminação e outros fatores do ambiente de observação, além de apresentar um alto custo computacional para criar e manter um modelo de fundo.

Estimativa de fundos

A criação de um modelo de fundo confiável é fundamental para o sucesso dos métodos dessa abordagem. O método mais simples para estimativa de fundo é obter

uma imagem do ambiente de observação sem a presença de objetos em movimento [16]. Porém, esta solução é impraticável em ambientes não controlados como é o caso de vias públicas de grande movimento. Como alternativa, pode-se utilizar um conjunto de quadros iniciais da sequência de imagem e aplicar uma medida estatística (média ou mediana) para estimar o fundo. A aplicação da média é simples e eficiente [17], porém, muito influenciada por valores extremos e discrepantes, no caso da mediana é uma estatística mais robusta não sendo influenciada por extremos, contudo implica em maior custo computacional e de armazenamento [2].

Existem outras abordagens que não necessitam de uma etapa de treinamento, ou seja, o modelo de fundo não é baseado em um conjunto de quadros iniciais. Nestas abordagens, o primeiro quadro da sequência é atribuído como modelo de fundo e a cada novo quadro este modelo é atualizado, assim, mudanças na iluminação e no ambiente são incorporadas ao modelo de fundo atualizado [15].

Algoritmos de Subtração de fundo

- Subtração de Fundo Estimado pela Média:

Este algoritmo utiliza os m quadros iniciais de treinamento para estimar o fundo a partir da média. O modelo $B(x,y)$ é estimado para cada pixel (x,y) como:

$$B(x,y) = \frac{1}{m} \sum_{i=0}^m f_i(x,y), \quad (2.1)$$

onde m é o número de quadros iniciais da sequência, com $1 < m < n$, n é o número total de quadros e $f_i(x,y)$ é a imagem correspondente ao quadro i da sequência.

O resultado do processo de segmentação para um quadro i de entrada, $A_i(x,y)$, é calculado como a diferença simétrica entre o quadro $f_i(x,y)$ e o modelo de fundo estimado $B(x,y)$. O retorno para cada pixel de $A_i(x,y)$ será 1 (um), se a diferença simétrica for maior que um limiar previamente definido τ , ou 0 (zero), caso contrário, segundo equação (2.2).

$$A_i(x, y) = \begin{cases} 1, & \text{se } |f_i(x, y) - B(x, y)| > \tau; \\ 0, & \text{caso contrário.} \end{cases} \quad (2.2)$$

Este procedimento é relativamente rápido, porém o resultado do processo de segmentação depende muito do limiar τ escolhido. Além disso, um mesmo limiar pode apresentar resultados bons para uma dada sequência de imagens e ruins para outra. Outro problema desta abordagem é a falta de adaptação do fundo, visto que, uma vez estimado, o fundo não é mais atualizado. Assim, um objeto do cenário que foi movido depois que o fundo foi estimado será reconhecido como um falso positivo.

- Subtração de Fundo Estimado pela Mediana:

Este algoritmo é um pouco mais robusto se comparado ao de médias, visto que não apresenta sensibilidade a valores extremos e discrepantes. A estimativa do fundo ($B(x, y)$) é calculada a partir da mediana dos m primeiros quadros da sequência de imagens de acordo com a equação abaixo:

$$B(x, y) = \text{mediana}_{\{1 < i < m\}} f_i(x, y), \quad (2.3)$$

onde m é o número de quadros iniciais da sequência, com $1 < m < n$, n é o número total de quadros e $f_i(x, y)$ é a imagem correspondente ao quadro i da sequência

Assim como o algoritmo de Subtração de Fundo Estimado pela Média, o cálculo da segmentação $A_i(x, y)$ é obtido utilizando a equação (2.2) e tem também como problema a falta de atualização do fundo estimado.

- Subtração de Fundo Estimado pela Média Adaptativa:

Este algoritmo diferencia-se dos listados acima por apresentar uma maior robustez à alteração de iluminação e alterações no cenário. Ele é também uma alternativa a situações onde não se tem um conjunto inicial de quadros sem objetos alvos, ou seja, quando o cenário de interesse encontra-se em constante movimento dificultando a criação de um modelo de fundo estático.

Para estimação do fundo, adota-se a imagem inicial da sequência como sendo um fundo estimado. A cada novo quadro de entrada, o modelo de fundo é atualizado, utiliza-se para isso uma taxa de aprendizado $\alpha \in (0,1)$, essa taxa é o que

indica o quanto do modelo atual i será utilizado para estimação do novo modelo de fundo. O modelo de fundo $B_i(x, y)$ é obtido pela equação seguinte:

$$B_i(x, y) = \begin{cases} f_{i-1}(x, y), & \text{se } i = 2; \\ (1 - \alpha)B_{i-1}(x, y) + \alpha f_{i-1}(x, y), & \text{se } i > 2. \end{cases} \quad (2.4)$$

O resultado do processo de segmentação para um quadro i de entrada, $A_i(x, y)$, é calculado como a diferença simétrica entre o quadro $f_i(x, y)$ e o modelo de fundo atualizado $B_i(x, y)$. O retorno para cada pixel de $A_i(x, y)$ será 1 (um), se a diferença simétrica for maior ou igual a um limiar previamente definido τ , ou 0 (zero), caso contrário, de acordo com a próxima equação.

$$A_i(x, y) = \begin{cases} 1, & \text{se } |f_i(x, y) - B_i(x, y)| \geq \tau; \\ 0, & \text{caso contrário;} \end{cases} \quad (2.5)$$

A maior dificuldade na utilização deste algoritmo é a escolha da taxa de aprendizado α [17]. Caso α seja muito perto de 0 (zero), o fundo irá se adaptar muito lentamente às alterações no cenário, o que pode gerar falsos positivos por um longo período, visto que o fundo atual pode não corresponder ao cenário modificado. Entretanto, caso α seja muito próximo de 1 (um), o fundo irá adapta-se muito rapidamente ao cenário e partes dos alvos podem ser perdidas pois serão rapidamente considerados parte do fundo.

- Subtração de Fundo Estimado pela Mediana de um intervalo:

Para evitar o problema da escolha da taxa de aprendizagem, pode-se utilizar a mediana dos últimos m quadros da sequência de imagens para estimar o fundo. O fundo para o quadro i , $B_i(x, y)$, é obtido como:

$$B_i(x, y) = \begin{cases} \text{mediana}_{\{1 \leq j < i\}} f_j(x, y), & \text{se } i - m < 1; \\ \text{mediana}_{\{i-m \leq j < i\}} f_j(x, y), & \text{se } i - m \geq 1; \end{cases} \quad (2.6)$$

onde $m \in (1, n)$, n é o numero total de quadros da sequência.

O cálculo da segmentação $A_i(x, y)$, é obtido utilizando a equação (2.5). Este algoritmo é bastante robusto e fácil de parametrizar, visto que não possui a taxa de

aprendizado. No entanto, é pouco mais lento e exige mais poder computacional, pois necessita do cálculo da mediana a cada iteração.

2.2.2 Diferença Temporal

Assim como na Subtração de Imagens de Fundos (seção 2.2.1), a técnica de diferença temporal é bastante simples [18]. Consiste na diferença simétrica de duas imagens consecutivas aplicadas a um limiar de classificação τ previamente definido [19]. Ela é calculada de acordo com a equação (2.7), tendo como $\delta_n(x, y)$ o módulo da diferença das imagens $f_n(x, y)$ e $f_{n-1}(x, y)$, sendo n o tempo atual.

$$\delta_n(x, y) = |f_n(x, y) - f_{n-1}(x, y)|. \quad (2.7)$$

O resultado do processo de segmentação para um quadro i de entrada, $A_i(x, y)$, é obtido segundo equação abaixo.

$$A_i(x, y) = \begin{cases} 1, & \text{se } \delta_i(x, y) \geq \tau; \\ 0, & \text{caso contrário;} \end{cases} \quad (2.8)$$

A diferença temporal é bastante eficiente computacionalmente, além de apresentar robustez a alterações de iluminação, ruídos e a movimentos periódicos da região de fundo, portanto pode detectar com precisão os objetos de interesses em circunstâncias variadas [1]. Collins *et al.* [19] afirmam que modelos baseados em diferença temporal são eficazes na detecção de pequenas variações de cena. Entretanto, a segmentação dos objetos alvos é incompleta, contendo espaços vazios no interior dos mesmos. Como alternativa, pode-se adotar um método híbrido onde primeiramente é utilizada a diferença temporal para identificar regiões de movimento e em seguida a subtração de imagens de fundo [19].

2.2.3 Modelos estatísticos

Os modelos estatísticos vêm se tornando bastante populares [2], pois possuem a capacidade de lidar com fundos dinâmicos e alterações de ambientes. Para cada pixel do modelo de fundo é computada uma ou mais grandezas estatísticas que são utilizadas para classificar os pixels de um novo quadro de entrada como *foreground* ou *background*.

Atualmente, existem várias técnicas estatísticas para se computar o modelo de fundo de uma sequência de imagens [18]. Uma abordagem mais detalhada sobre essa diversidade de técnica é feita em Lara [20]. Neste trabalho, descreve-se o funcionamento do modelo de mistura de gaussianas proposto por Stauffer e Grimson [21].

Modelo de Mistura de Gaussianas

Nesta abordagem, cada pixel é representado por k funções gaussianas (tipicamente com k entre 3 e 5) [22]. Se um pixel do quadro de entrada não se ajusta às respectivas gaussianas que modelam o fundo da imagem são considerados partes do objeto alvo.

A etapa de atualização do modelo de gaussianas acontece após o processamento do quadro de entrada. Nessa etapa, é realizada uma avaliação da gaussiana mais representativa para o pixel, dentre as k gaussianas do modelo, e esta tem a média e a variância atualizada. Caso nenhuma das gaussianas represente corretamente o pixel, uma nova função gaussiana é adicionada à mistura em substituição da menos representativa.

A equação (2.9) apresenta como cada pixel (x, y) do quadro i tem seus valores modelados por uma mistura de k funções gaussianas. Onde $P_i(x, y)$ representa a probabilidade do pixel (x, y) , k é o número de gaussianas utilizadas, $\omega_{j,i}(x, y)$ é uma ponderação para a j -ésima gaussiana, $\mu_{j,i}(x, y)$ e $K_{j,i}(x, y)$ são, respectivamente, o valor médio e a matriz de covariância da j -ésima gaussiana e $\eta\{f_i(x, y), \mu_{j,i}(x, y), K_{j,i}(x, y)\}$ é a função componente (densidade de probabilidade) que é definida segundo a equação **Erro! Fonte de referência não encontrada.**:

$$P_i(x, y) = \sum_{j=1}^k \omega_{j,i}(x, y) \eta\{f_i(x, y), \mu_{j,i}(x, y), K_{j,i}(x, y)\}, \quad (2.9)$$

$$\eta\{f_i(x, y), \mu_{j,i}(x, y), K_{j,i}(x, y)\} = \frac{1}{(2\pi)^{\frac{n}{2}} |K_{j,i}(x, y)|^{\frac{1}{2}}} e^{-\frac{1}{2} W_i}, \quad (2.10)$$

$$W_i = \{f_i(x, y) - \mu_{j,i}(x, y)\}^T K_{j,i}(x, y)^{-1} \{f_i(x, y) - \mu_{j,i}(x, y)\}, \quad (2.10.1)$$

onde $f_i(x, y)$ é o valor do pixel (x, y) no quadro i e $T \in (0,1)$.

Visando diminuir o custo computacional, assume-se que a matriz de covariância é definida como:

$$K_{j,i} = \sigma_{j,i}^2 f_i, \quad (2.11)$$

onde $\sigma_{j,i}^2$ é o desvio-padrão da j -ésima gaussiana calculada até o quadro i .

A equação (2.12) é utilizada para atualização dos pesos das k distribuições gaussianas, onde $\alpha \in (0,1)$ é uma taxa de aprendizado, $M_{j,i}$ vale "1" para a distribuição que casou com o pixel e 0 caso contrário, $\omega_{j,i}(x, y)$ é uma ponderação para a j -ésima gaussiana.

$$\omega_{j,i} = (1 - \alpha)\omega_{j,i-1} + \alpha(M_{j,i}). \quad (2.12)$$

A média e o desvio-padrão da distribuição que casou com o valor do pixel é atualizado segundo as equações abaixo:

$$\mu_i = (1 - \rho)\mu_{i-1} + \rho f_i(x, y); \quad (2.13)$$

$$\sigma_i^2 = (1 - \rho)\sigma_{i-1}^2 + \rho\{f_i(x, y) - \mu_i\}^T \{f_i(x, y) - \mu_i\}, \quad (2.14)$$

onde ρ é uma taxa de aprendizado para a atualização da média e do desvio-padrão.

As k distribuições que representam um determinado pixel são ordenadas de acordo com a relação $\frac{\mu_i}{\sigma_i}$. As B_i primeiras distribuições são utilizadas como modelo de *background* do pixel para a cena. O valor de B_i é dado pela equação (3.4).

$$B_i = \operatorname{argmin}_b \left\{ \sum_{j=1}^b \omega_{j,i} > T \right\}. \quad (2.15)$$

2.3 Pós-Segmentação

Modelos de segmentação de objetos geralmente estão sujeitos a falhas no processo de segmentação devido à existência de sombras, alterações na iluminação do ambiente, movimentos de folhas e galhos de árvores causados pelo vento, além de ruídos originados pela obtenção de imagens das câmeras digitais. Desta forma, é necessária, após a aplicação dos métodos de segmentação de objetos, uma etapa de pós-segmentação que é caracterizada principalmente pela eliminação de ruídos, agrupamentos de objetos em regiões semelhantes e pela identificação das áreas de interesses para os sistemas de rastreamento de objetos [14].

A **Figura 2 - 3** apresenta um exemplo de segmentação de objetos em um ambiente externo sujeito à alteração da iluminação.

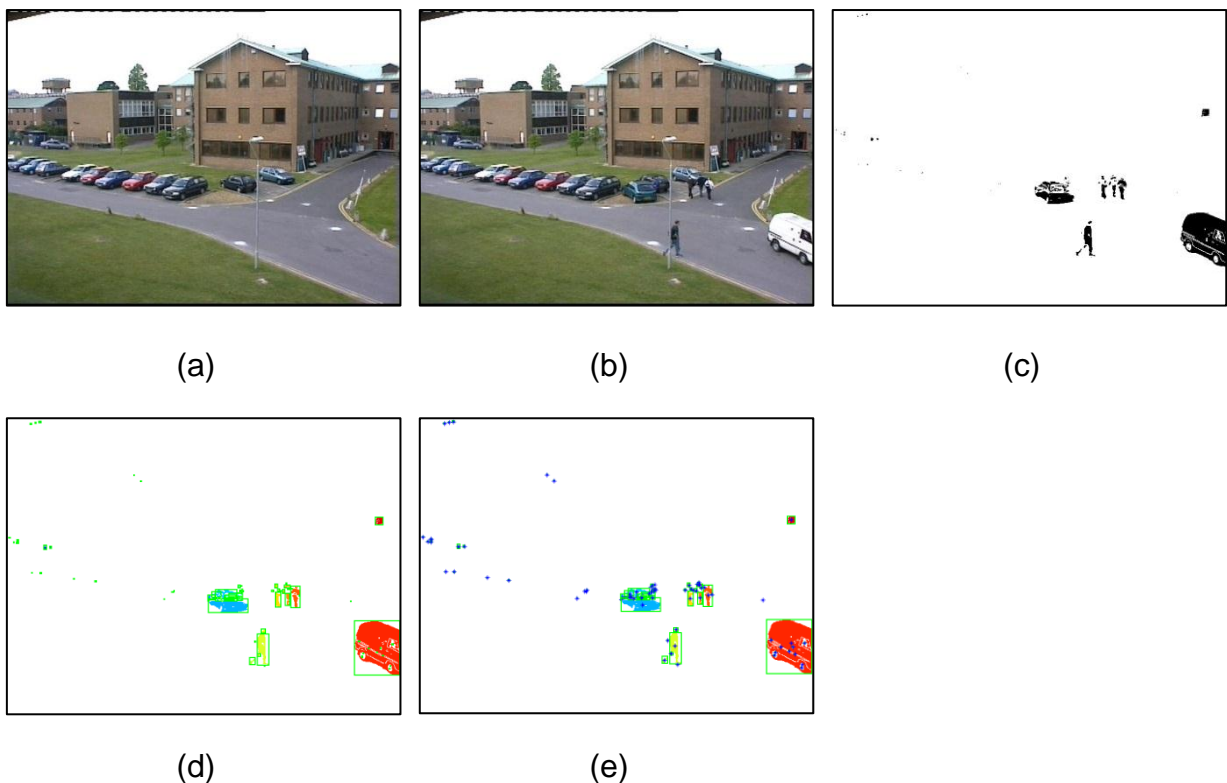


Figura 2 - 3. Ruídos na segmentação causados por alteração na iluminação. (a) Modelo de fundo do ambiente, (b) Quadro de entrada para segmentação, (c) Quadro resultante da segmentação, (d) Identificação de regiões - retângulos mínimos, (e) Identificação de regiões - retângulos mínimos e centróide.

Como pode ser observado na **Figura 2 - 3** (c), a imagem apresenta ruídos no processo de segmentação que irão afetar na etapa de identificação de objetos de

interesses **Figura 2 - 3** (d) (e). Nas próximas subseções serão apresentadas algumas estratégias utilizadas para melhorar os resultados dos métodos de segmentação, mais informações podem ser obtidas em [14].

2.3.1 Identificação de Regiões

A identificação de regiões visa separar objetos alvos do restante da imagem. A identificação das regiões é feita utilizando algoritmos de detecção de regiões conectadas [14] que são responsáveis por calcular retângulos mínimos contendo os objetos de interesses. Esses retângulos computados serão posteriormente utilizados para a extração de características dos objetos pelos algoritmos de rastreamento. A **Figura 2 - 4** mostra um exemplo de identificação de regiões.



Figura 2 - 4. Exemplo de identificação de regiões utilizando a representação de retângulos mínimos e centróide.

2.3.2 Eliminação de Ruídos

Após a segmentação de objetos, as imagens resultantes podem apresentar falsos positivos provenientes do processo de segmentação. Falsos positivos dessa natureza podem ser removidos utilizando operações primitivas no processamento de imagens, como é o caso de filtro de mediana, filtragem no domínio da frequência, erosão, dilatação, abertura, fechamento, preenchimento morfológico de buracos [14].

A eliminação de ruídos não garante bons resultados para qualquer situação. Por exemplo, a aplicação de erosão e dilatação pode acarretar na eliminação de pequenos objetos de interesses, principalmente em ambientes externos onde a tomada da cena abrange uma área maior e sofre mais influências de fatores

ambientais (iluminação, movimentos de galhos causados pelo vento, entre outros). Além disso, pode ocorrer a junção de objetos pertencentes a regiões diferentes.

A **Figura 2 - 5** apresenta o processo de eliminação de ruídos; a **Figura 2 - 5 (a)** apresenta a segmentação resultante da aplicação do algoritmo de subtração de fundo (ver seção 2.2.1); a **Figura 2 - 5 (b)** apresenta o resultado após a aplicação de fechamento de regiões [14]; a **Figura 2 - 5 (c)** identifica as regiões conectadas, a cor verde representa áreas que satisfazem condições desejadas, (neste exemplo foi utilizado como critério a área mínima), as de vermelho são ruídos identificados após avaliação da condição estabelecida; **Figura 2 - 5 (d)** apresenta o resultado após a eliminação de pequenas regiões; **Figura 2 - 5 (e)** identifica as regiões de interesses e traça os retângulos mínimos e o centróide; **Figura 2 - 5 (f)** apresenta o resultado final do processo de eliminação de ruídos.

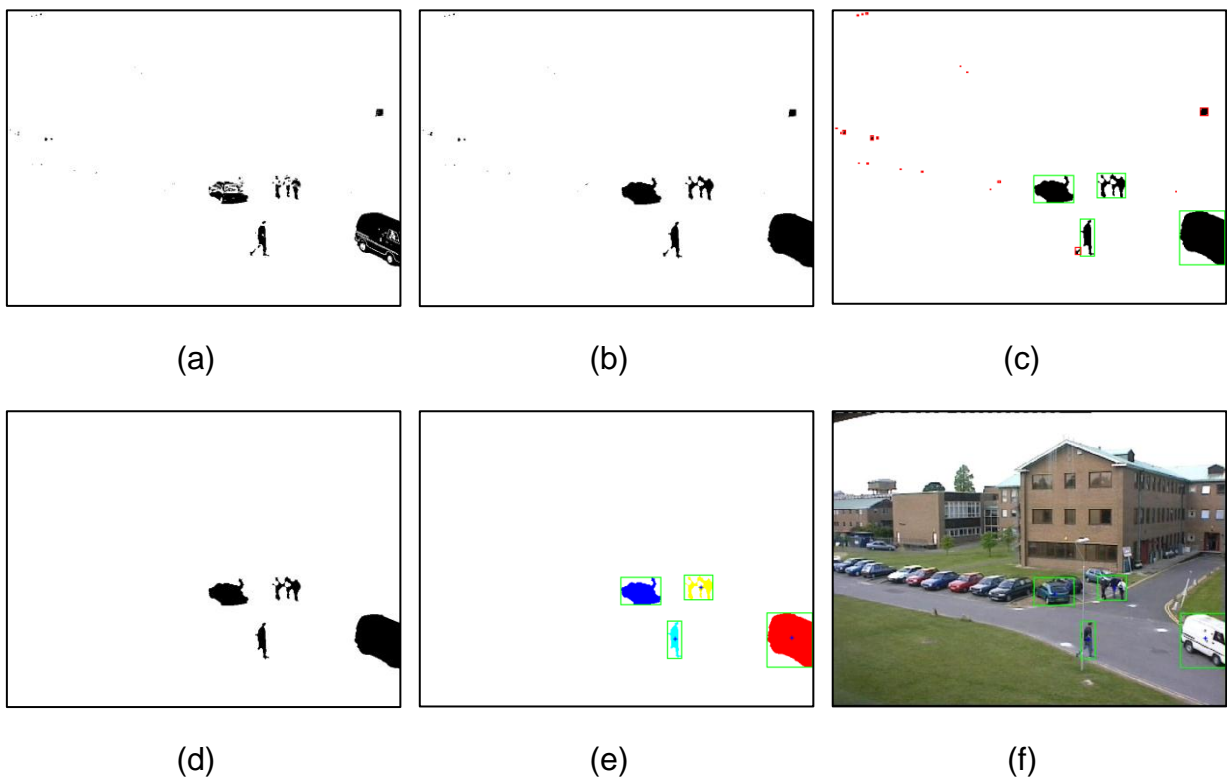


Figura 2 - 5. Exemplo de eliminação de ruídos. (a) segmentação inicial, (b) fechamento de regiões, (c) identificação de regiões conectadas (em vermelho as regiões que serão descartadas), (d) Resultado da eliminação de regiões pequenas, (e) Identificação de regiões de interesses, (f) resultado final do processo de eliminação de ruídos.

Capítulo 3

Rastreamento de Objetos

Após a etapa de segmentação de movimento (Capítulo 2), os objetos alvos da cena (*foreground*) encontram-se separados dos elementos estáticos (*background*) e estão prontos para a etapa de rastreamento de objetos. O objetivo do rastreamento de objetos é gerar a trajetória dos objetos alvos ao longo do tempo, localizando sua posição em cada quadro do vídeo, ou seja, identificar no quadro atual de um vídeo, a correspondência com objetos alvos detectados no quadro anterior e estabelecer uma relação temporal entre os objetos alvos detectados.

Neste capítulo serão apresentadas as principais técnicas para rastreamento de objetos por similaridade de características. A seção 3.1 apresenta as abordagens utilizadas para estabelecer a correspondência temporal entre objetos em quadros de vídeos. Na seção 3.2 são apresentados possíveis casos de correspondência entre objetos em quadros consecutivos. Por fim, as seções 3.3 a 3.5 apresentam possíveis estados assumidos pelos objetos durante o processo de rastreamento e técnicas de detecção e tratamento de oclusão/divisão de objetos.

3.1 Rastreamento de objetos por similaridades de características

O rastreamento de objetos por similaridades de características consiste em estabelecer relações temporais dos objetos alvos utilizando para isso características de baixo nível facilmente obtidas dos objetos segmentados na fase de segmentação de movimento.

No trabalho desenvolvido por Amer [3], o monitoramento de objetos com base na similaridade de características é feito em três etapas: extração de características, correspondência de objeto, e monitoramento de características. Na primeira etapa, objetos são segmentados e as suas características espaciais e temporais são computadas (A **Tabela 3 - 1** apresenta algumas das características utilizadas por Amer). A saída desta etapa é $O(n)$ e $O(n - 1)$, onde $O(n)$ é a lista de objetos no

quadro $I(n)$ e $O(n - 1)$ a lista de objetos no quadro $I(n - 1)$, sendo $I(n)$ a função que representa a imagem, e n o instante de tempo.

Tabela 3 - 1. Características utilizadas por Amer [8].

Nome	Definição	Descrição
Altura	$h = y_{max} - y_{min}$	y_{max} : maior valor de y assumido por um <i>pixel</i> do objeto; y_{min} : menor valor de y assumido por um <i>pixel</i> do objeto.
Largura	$w = x_{max} - x_{min}$	x_{max} : maior valor de x assumido por um <i>pixel</i> do objeto; x_{min} : menor valor de x assumido por um <i>pixel</i> do objeto.
Área	a	Número de <i>pixel</i> do objeto.
Perímetro	p	Número de <i>pixel</i> na borda do objeto.
Retângulo Mínimo	$MBB = (x_{min}, x_{max}, y_{min}, y_{max})$	Idem descrição de Altura e Largura.
Compactação	$e = \frac{h}{w}$ se $h < w$ ou $e = \frac{w}{h}$ se $w < h$	h : altura ; w : largura.
Irregularidade	$r = \frac{p^2}{4\pi a}$	p : perímetro; π : 3,14; a : área.

Na segunda etapa, cada objeto do quadro anterior é comparado com um objeto do quadro atual criando uma única correspondência entre eles. Esta etapa é dividida em duas fases. Na primeira fase, é criada uma correspondência para cada objeto $O_{lp} \in I(n - 1)$ com pelo menos um objeto $O_{ic} \in I(n)$, representando esta correspondência por $M_{li} : O_{lp} \rightarrow O_{ic}$. A correspondência na primeira etapa do algoritmo proposto por Amer [3] é feita de acordo com a seguinte equação:

$$M_{li} : (d_{li} < t_r) \wedge (\zeta_{li} > t_v) \wedge (|w_{xic}| < w_{max}) \wedge (|w_{yic}| < w_{max}), \quad (3.1)$$

\bar{M}_{li} : Caso contrário,

onde M_{li} é uma função booleana que retorna verdadeiro quando uma correspondência é encontrada entre $O_{lp} \in I(n - 1)$ e $O_{ic} \in I(n)$, \bar{M}_{li} significa que não houve correspondência entre os objetos. t_r , t_v e w_{max} são limiares aplicados, d_{li}

é a distância calculada entre os objetos O_{lp} e O_{ic} , ζ_{li} representa um valor resultante das funções de votação.

Na segunda fase desta etapa, se O_{lp} possui mais de uma correspondência com objetos de $I(n)$, então passa por um processo de votação majoritária para eliminar ambiguidades. Assume-se que, objetos O_{ic} que não possuem nenhuma correspondência, são objetos que acabaram de entrar na cena, ou seja, objetos criados, e objetos O_{lp} que não possuem correspondência são considerados objetos que saíram da cena. A **Figura 3 - 1** apresenta o fluxo da segunda etapa do algoritmo proposto por Amer [3].

Na terceira etapa, são monitorados problemas como alterações nos objetos identificados, oclusões ou divisão de objetos, e as características dos objetos são corrigidas. Essas novas características são utilizadas para atualizar os resultados obtidos nas etapas anteriores do processo.

Uma abordagem mais simples é proposta por Dedeoglu [4]. Ele utiliza para realizar a correspondência, apenas a área dos objetos e o centro de massa. A área dos objetos é calculada por $s = w * h$, onde w é a largura do objeto e h é a altura; o centro de massa (x_c, y_c) é calculado por $x_c = \frac{1}{n} \sum_{i=0}^n x_i$; $y_c = \frac{1}{n} \sum_{i=0}^n y_i$, onde x_c é a coordenada x do centro de massa, y_c é a coordenada y do centro de massa do objeto, n é o número de pixels do objeto, x_i e y_i são as coordenadas do i -ésimo pixel do objeto.

Em sua abordagem, Dedeoglu [4] utiliza um grafo bipartido $G(m, n)$ onde os vértices representam os objetos (uma partição de vértice representam objetos anteriores O_{lp} e a outra partição representa novos objetos O_{ic}) e as arestas representam uma correspondência entre dois objetos, com m sendo a quantidade de objetos no quadro anterior e n a quantidade de objetos no quadro atual.

A correspondência entre os objetos O_{lp} e O_{ic} é calculada análoga ao método propostos por Amer, ou seja, para cada objeto O_{lp} verifica-se a existência de uma correspondência em O_{ic} .

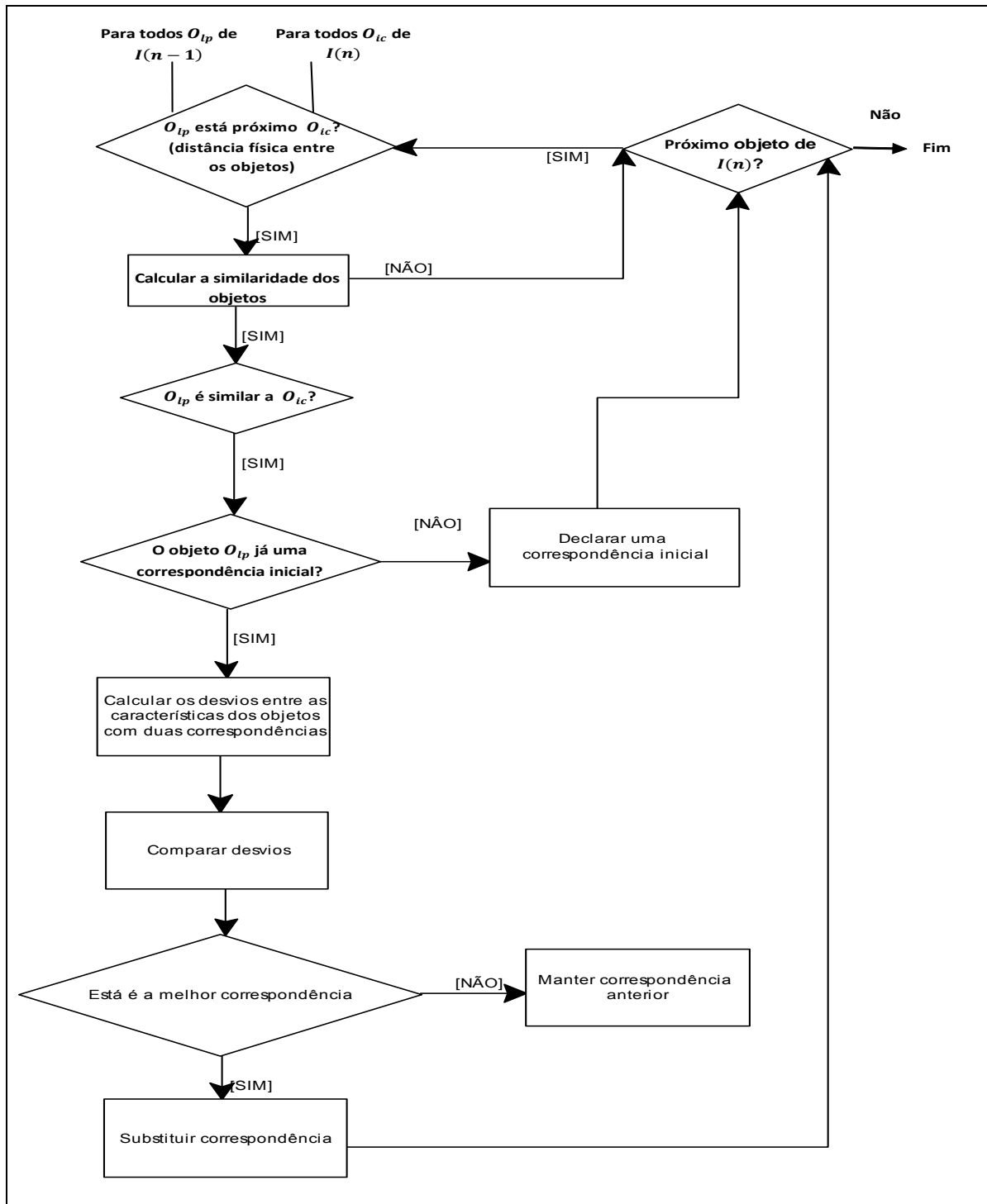


Figura 3 - 1. Fases da etapa de casamento de características [3].

O critério para realizar a correspondência entre os objetos é definido pela distância entre os centros de massas dos dois objetos aplicada a um limiar t pré-definido, como é mostrado abaixo:

$$Dist(c_{tp}, c_{ic}) < t, \tag{3.2}$$

onde $Dist(. , .)$ é a distância euclidiana entre os dois pontos que é:

$$Dist(c_{lp}, c_{ic}) = \sqrt{(x_{c_{lp}} - x_{c_{ic}})^2 + (y_{c_{lp}} - y_{c_{ic}})^2}. \quad (3.3)$$

Depois do cálculo das distâncias entre os objetos, uma nova verificação é feita levando em consideração a área dos objetos, S_i e S_p representam a área dos objetos O_{ic} e O_{lp} respectivamente. Esta segunda verificação é definida por

$$\frac{S_p}{S_i} < \mu \text{ ou } \frac{S_i}{S_p} < \mu, \quad (3.4)$$

onde μ é uma constante pré-definida representando um limite para a equação acima. Esta verificação é bastante útil, pois elimina a possibilidade de corresponder uma região pequena com uma região grande causada por segmentações imprecisas.

Uma questão bastante relevante na escolha dos algoritmos de rastreamento por similaridades de características é a quantidade de características que serão utilizadas para realizar a correspondência entre os objetos. Um algoritmo com poucas características, como é o caso do proposto por Dedeoglu [4], é fácil de implementar e apresenta bom desempenho computacional, entretanto, algoritmos com um maior número de características (Amer [3]) apresentam melhores resultados na correspondência entre os diversos objetos da cena.

3.2 Correspondência entre Objetos

Ao estabelecer uma correspondência entre os objetos encontrados no quadro anterior ($O(n-1)$) e os objetos encontrados no quadro corrente ($O(n)$) podem ocorrer cinco casos diferentes de correspondência [4].

1. Um-para-Um: Um objeto do quadro anterior apresenta correspondência com um objeto do quadro corrente (**Figura 3 - 2**). As características do objeto O_{lp} são atualizadas com as informações recebidas de O_{ic} .

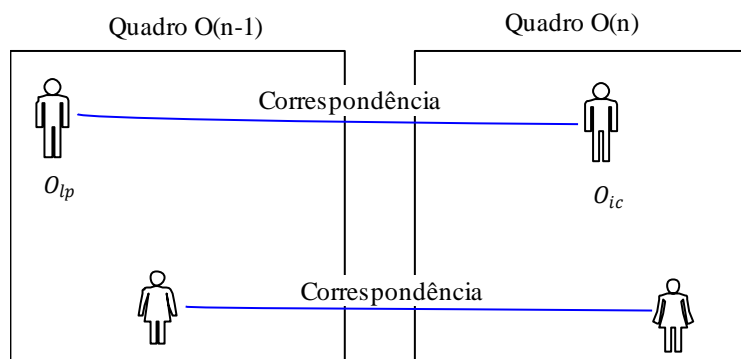


Figura 3 - 2. Correspondência Um-para-Um.

2. Um-para-Muitos: Um objeto do quadro anterior apresenta correspondência com mais de um objeto do quadro corrente (**Figura 3 - 3**). Este caso pode ocorrer devido à divisão de objetos (*splitting*) causada por problema na segmentação, ou por objetos muito próximos e com características similares. Objetos neste estado devem passar por um novo processamento, onde outras características dos objetos são comparadas e são reduzidos ao caso 1.

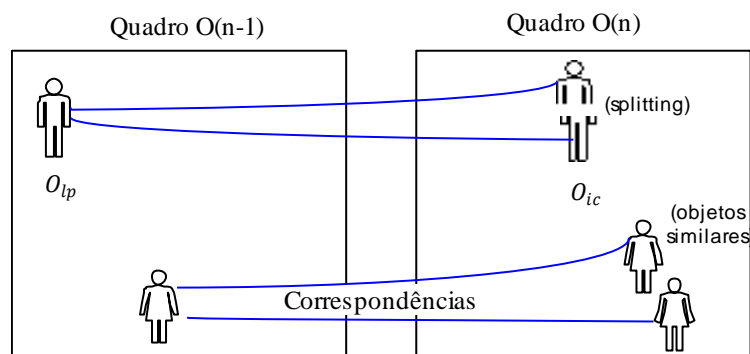


Figura 3 - 3. Correspondência Um-para-Muitos.

3. Um-para-Nenhum: Um objeto do quadro anterior não apresenta correspondência com objeto do quadro corrente (**Figura 3 - 4**). Este caso ocorre quando um objeto desaparece da cena, isto é, quando um objeto sai do limite de captura da câmera, ou se ocorrer oclusão entre objetos. No caso de oclusão, o objeto é preservado até a etapa de detecção de oclusão. Caso contrário, esse objeto é excluído da lista de objetos do quadro anterior.

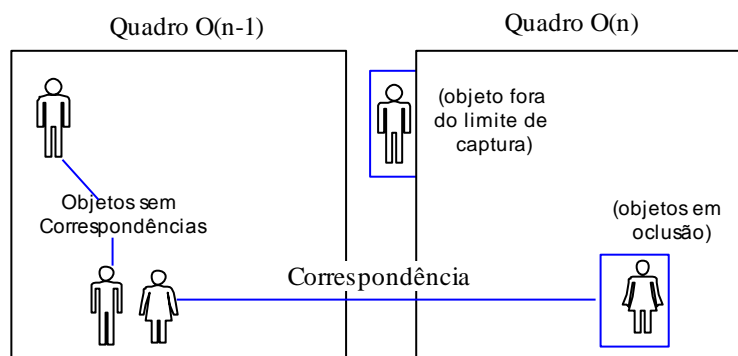


Figura 3 - 4. Correspondência Um-para-Nenhum.

4. Nenhum-para-Um: Não existe correspondência na lista de objetos do quadro anterior para um objeto do quadro corrente (**Figura 3 - 5**). Este caso ocorre quando um novo objeto entra na cena, ou ocorrer a separação de objetos em oclusão. Caso seja um novo objeto, este é adicionado à lista de objetos em rastreamento. Se isso é devido à separação de objetos em oclusão, a correspondência será dada por um procedimento de tratamento de objetos em oclusão.

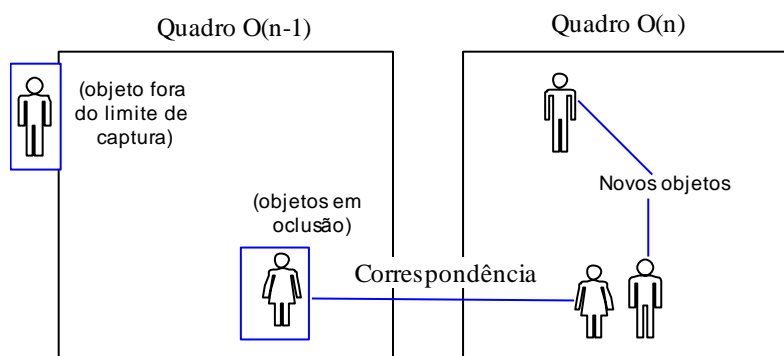


Figura 3 - 5. Correspondência Nenhum-para-Um.

5. Muitos-para-Um: Alguns objetos do quadro anterior apresentam correspondência com um mesmo objeto do quadro corrente (**Figura 3 - 6**). Este caso pode ocorrer devido à oclusão de objetos. Objetos neste estado devem passar por um novo processamento, onde outras características dos objetos são comparadas e são reduzidos ao caso 1.

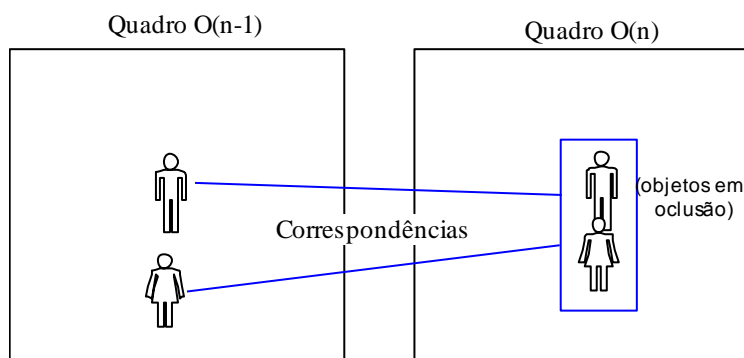


Figura 3 - 6. Correspondência Muitos-para-Um.

3.3 Estados dos Objetos

Como mostrado na seção anterior, objetos podem apresentar diferentes tipos de correspondências. Desta forma, para um rastreamento consistente é necessário criar estruturas que permitam rastrear corretamente objetos em suas possíveis mudança de correspondência. Com o objetivo de monitorar tais mudanças sem perder o objeto seguido ou fazer falsas afirmações, bem como evitar o rastreamento de ruídos, Lei e Xu [23] criaram um conjunto de sete estados (novo, rastreando, temporariamente indisponível, ocluído, perdido, dividido e fora de cena) para representar os possíveis estados assumidos por um objeto rastreado. O diagrama de transição de estado é apresentado na **Figura 3 - 7**.

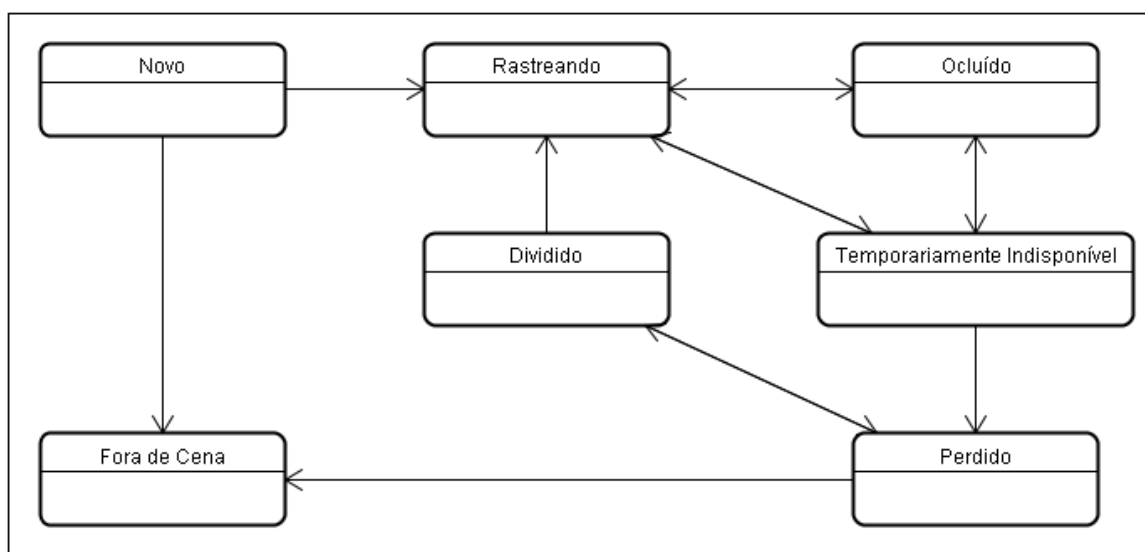


Figura 3 - 7. Diagrama de transição de estados assumidos por objetos em rastreamento [23].

A seguir, é apresentada uma descrição desses diferentes estados adotados por Lei e Xu [23]:

- **Novo:** é atribuído quando um novo objeto aparece na cena. Para transitar para o estado de "rastreamento", utilizam-se duas medidas de confiança (quantidade de quadros em que o objeto foi seguido e um "fator de movimento") [23], visando garantir que esse novo objeto é realmente um alvo em movimento. Caso contrário, será removido da fila de objetos e atribuído como "fora de cena";
- **Rastreamento:** é atribuído para objetos alvos que estão em movimento. Objetos neste estado podem sofrer oclusão, agrupando-se a outros objetos da cena ou causada por ruídos na segmentação, passando para o estado de "ocluído". Além disso, podem mudar para o estado "temporariamente indisponível", quando parte da estrutura da cena esconde o objeto (ex.: um poste no meio do caminho);
- **Ocluído:** é atribuído quando um objeto funde-se a outros objetos em movimento. Quando for identificada novamente a correspondência do objeto em oclusão, ele terá seu estado modificado para "rastreamento". Caso contrário, se depois de uma quantidade de quadros não seja possível criar uma correspondência, o estado é modificado para "temporariamente indisponível";
- **Temporariamente Indisponível:** se um objeto neste estado encontrar uma correspondência novamente retorna ao estado "rastreamento". Caso contrário, se continuar neste estado por um determinado período, será considerado como "perdido";
- **Perdido:** uma vez classificado como perdido, aplica-se as métricas de confiança para verificar se o objeto pode ser considerado como "dividido", ou seja, o objeto foi separado em duas ou mais partes (ex.: grupo de pessoas que se desfaz). Caso contrário, será considerado como "fora de cena";
- **Dividido:** um objeto entra neste estado quando verificado que ocorreu divisão deste em partes menores. Caso essas partes correspondam a

um mesmo objeto, ele permanecerá neste estado até que elas se unam novamente e depois passará para o estado "rastreamento". Caso contrário, se essas partes corresponderem a novos objetos em movimento será atribuído novos rótulos a estes objetos e passaram para o estado "rastreamento";

- **Fora de Cena:** objetos entram nesse estado quando ficaram no estado "perdido" por certo período de tempo, ou quando foram considerados como ruídos no estado "novo".

3.4 Oclusões de Objetos (*Occlusion*)

Um dos grandes problemas que interferem diretamente no desempenho dos algoritmos de rastreamento é a oclusão de objetos [3]. Oclusão ocorre quando dois ou mais objetos se sobrepõem e são considerados como fazendo parte de uma mesma estrutura (**Figura 3 - 4** e **Figura 3 - 6**).

Geralmente, o tratamento de oclusão envolve duas etapas. Na primeira etapa, a técnica de tratamento de oclusão deve ser capaz de detectar quando os objetos entram ou saem em situações de oclusão. Em seguida, a técnica deve ser capaz de rastrear os objetos enquanto estiverem ocluídos.

Segundo Dedeoglu [4], a detecção de oclusão pode ser alcançada verificando se uma região delimitadora de um novo objeto O_i se sobrepõe à região delimitadora de outros objetos do quadro anterior que não possuem correspondência no quadro corrente (caso 5 da seção 3.2). Depois de detectar a oclusão, não se exclui os objetos que foram sobrepostos, entretanto, o estado desses objetos são alterados conforme mostrado na seção 3.3.

3.5 Divisão de Objetos (*Splitting*)

Outro problema bastante comum no rastreamento de objetos é a divisão de objetos (**Figura 3 - 2** e **Figura 3 - 5**). Geralmente a divisão de objetos ocorre por problemas na segmentação (caso 2 da seção 3.2), onde um mesmo objeto é segmentado em diversas partes e cada parte é tida como sendo um novo objeto.

Entretanto, a divisão de objetos pode ocorrer quando objetos em estados de oclusão são separados (caso 4 da seção 3.2).

Para detectar a divisão de objetos, pode-se utilizar uma heurística semelhante à detecção de oclusão, ou seja, quando um objeto do quadro corrente é encontrado, verifica-se se houve objetos do quadro anterior que possuam sua região delimitadora sobrepondo o objeto encontrado. Caso isto ocorra, pode-se assumir que ocorreu a divisão de objetos em oclusão. No caso da divisão corresponder a partes de um mesmo objeto, adota-se a abordagem vista na seção 3.3 no estado "dividido".

Capítulo 4

Compressão e Indexação de Vídeo Baseado em Rastreamento de Objetos (CIVRO)

Neste capítulo, será apresentada uma estratégia para compressão de imagens para sistemas de rastreamentos de objetos visando economia do espaço utilizado para armazenamento e agilidade na recuperação de imagens de cenas específicas.

4.1 Visão Geral

Em geral, sistemas de monitoramento e rastreamento de objetos armazenam grande quantidade de vídeos digitais, pois, capturam imagens dos ambientes monitorados 24 horas por dia que devem ser arquivadas por um longo período de tempo, em virtude da necessidade de eventuais consultas.

Para entender melhor a quantidade de informação gerada por uma câmera ligada 24 horas com resolução de 720 x 480 x 24 bits e taxa de captura de 30 fps (quadros por segundo, *de quadros per second*), pode-se realizar o seguinte cálculo:

$$30 \frac{\text{frames}}{\text{s}} * (720 * 480) \frac{\text{pixel}}{\text{frames}} * 3 \frac{\text{bytes}}{\text{pixel}} * 3600 \frac{\text{s}}{\text{h}} * 24 \text{ h} \cong 2,68 * 10^{12} \text{ bytes.}$$
 Seriam necessários aproximadamente 2,44 TB (*terabytes*) para armazenar as imagens geradas por uma única câmera do ambiente monitorado sem nenhum tipo de compressão. O que tornaria inviável o monitoramento do ambiente.

Grande parte dos dados gerados por sistemas de monitoramento acaba ficando sem utilidade [10], pois não apresentam organização que permita uma recuperação eficiente, sendo armazenados apenas por critérios de segurança e de eventuais consultas. Isso gera uma grande dificuldade relacionada à recuperação de cenas específicas, por exemplo, na identificação da presença de pessoas não autorizadas em determinado ambiente, ou quando se deseja analisar apenas cenas

onde os objetos estão em movimento ou que possuam um determinado número de objetos em movimento com certas características. Seria necessário solicitar ao operador das câmeras o mapeamento manual de tais informações, desperdiçando bastante tempo com essa tarefa.

Existem algumas estratégias que minimizam alguns dos problemas acima elencados, como é o caso de compressão de vídeos [14] e modelos de geração semânticas para vídeos digitais [10, 24, 25].

Neste trabalho, é proposta uma abordagem de compressão de vídeo específica para sistemas de rastreamento de objetos visando um armazenamento eficiente das imagens obtidas por esses sistemas. O objetivo principal é eliminar informações desnecessárias das imagens obtidas, armazenando apenas informações de relevância para o rastreamento dos objetos alvos da aplicação, diminuindo informações redundantes (no sentido de não serem utilizadas).

Além disso, a abordagem proposta mantém informações de contexto das imagens armazenadas, dentre elas, informações sobre objetos segmentados e características extraídas pelo rastreamento de objetos. Esses metadados [26,27] ficam disponíveis para consultas posteriores, além de serem utilizados para realizar a indexação do vídeo gerado e auxiliam no processo de decodificação. A **Figura 4 - 1** apresenta o modelo do sistema proposto.



Figura 4 - 1. Modelo do sistema proposto.

De acordo com a **Figura 4 - 1**, inicialmente é feita a captura das imagens do ambiente (podendo ser: imagens sequenciais, vídeo armazenado ou vídeo em tempo real), que são passadas como entrada para o módulo de segmentação de movimento onde os objetos alvos da aplicação são identificados e segmentados. Os objetos identificados e segmentados são processados pelo módulo de rastreamento de objetos, cuja responsabilidade é extrair as características dos objetos e estabelecer a correspondência temporal entre os objetos encontrados. Finalmente, a saída do rastreamento é exibida e armazenadas de acordo com o modelo proposto através de uma etapa de codificação que será apresentada na próxima seção.

4.2 Compressão

O termo compressão refere-se ao processo de reduzir o volume de dados necessários para representar dada quantidade de informações [14]. A compressão de imagens visa diminuir o espaço necessário para armazenar, entretanto, devem-se manter as características originais da imagem não prejudicando a sua recuperação para a aplicação fim.

Em geral, os sistemas de compressão de imagens procuram eliminar três tipos de redundância:

- Redundância de codificação: refere-se à quantidade de bits utilizados para representar um pixel. Uma pequena quantidade de bits pode gerar imagens com pouca variação de cor, entretanto, quando o número de bits é alto, muita informação é necessária para o armazenamento e a variação de níveis de cor é quase que imperceptível;
- Redundância espacial e temporal: refere-se a informações replicadas de pixels vizinhos. Como geralmente pixels próximos apresentam uma intensidade similar, as informações são desnecessariamente replicadas nas representações dos pixels vizinhos;
- Informações irrelevantes: refere-se a informações ignoradas pelo sistema visual humano e/ou informações irrelevantes para utilização pretendida.

A abordagem desenvolvida preocupa-se em eliminar informações irrelevantes para os sistemas de rastreamento e monitoramento de objetos; não impedindo que sejam utilizadas outras técnicas de compressão em conjunto.

Em geral, um sistema de compressão é composto por dois componentes de funcionamento distintos: um codificador e um decodificador. O codificador realiza a compressão e o decodificador, a operação complementar de descompressão.

Na abordagem desenvolvida neste trabalho, o codificador é alimentado com objetos segmentados com características extraídas passadas pelo módulo de rastreamento de objetos (como foi visto na **Figura 4 - 1**) e um fundo de referência, tendo como saída uma representação comprimida da entrada. Essa representação é armazenada para posterior utilização. Quando esta representação comprimida é apresentada ao decodificador, a imagem original é reconstruída. A **Figura 4 - 2** mostra o diagrama de blocos do funcionamento do sistema de compressão proposto neste trabalho.

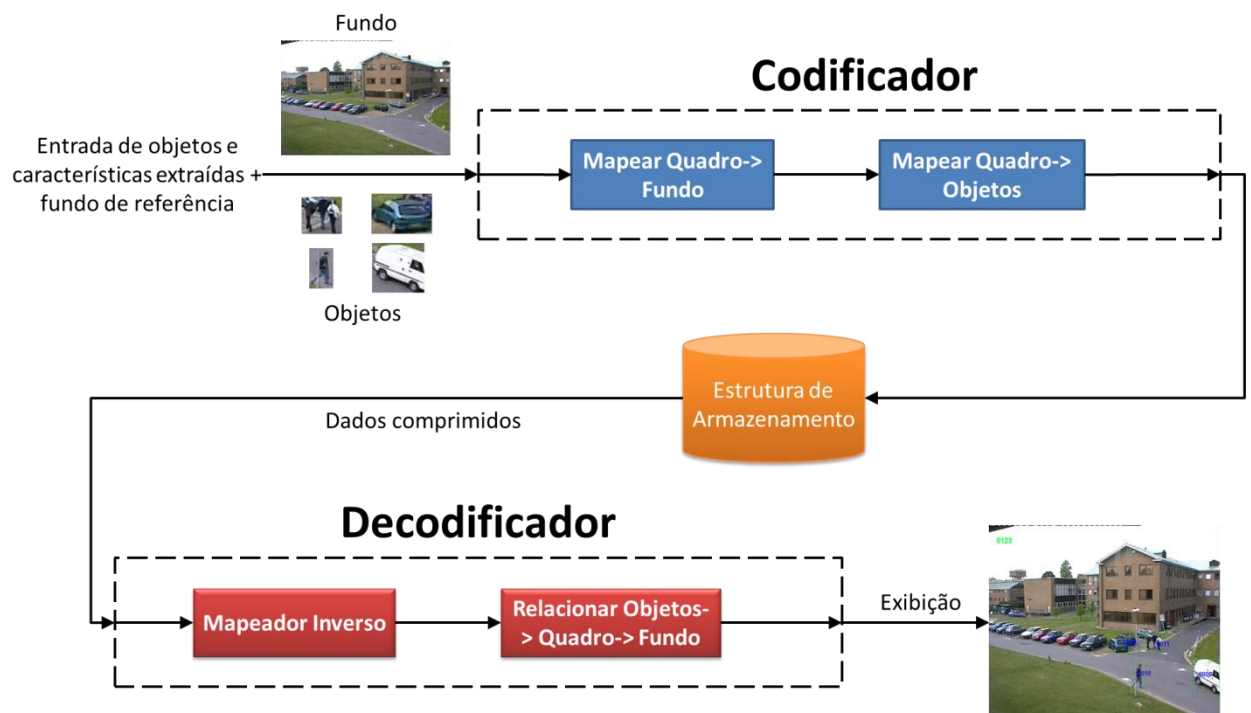


Figura 4 - 2. Diagrama de blocos do sistema de compressão.

A imagem reconstruída não é uma réplica exata da imagem de entrada, visto que muitas informações irrelevantes são eliminadas, caracterizando desta forma, um sistema de compressão com perda. É válido salientar que a compressão proposta,

depende do desempenho da técnica de rastreamento utilizada, pois as informações armazenadas são obtidas pelo módulo de rastreamento. Além disso, a abordagem é ideal para ambientes onde o fundo de referência apresenta baixa taxa de atualização.

4.2.1 Codificador

A codificação se inicia com a entrada dos dados pelo sistema de rastreamento de objetos. São passadas como parâmetros, informações sobre o fundo de referência, os objetos em cena e as características desses objetos que foram extraídas pelo módulo de rastreamento de objetos por similaridade de características (mais detalhes sobre essas características podem ser obtidas na **Tabela 3 - 1**). Para cada objeto de entrada é passada também sua posição na cena, ou seja, sua localização em relação à imagem de entrada. A **Figura 4 - 3** apresenta o diagrama de blocos do codificador proposto.

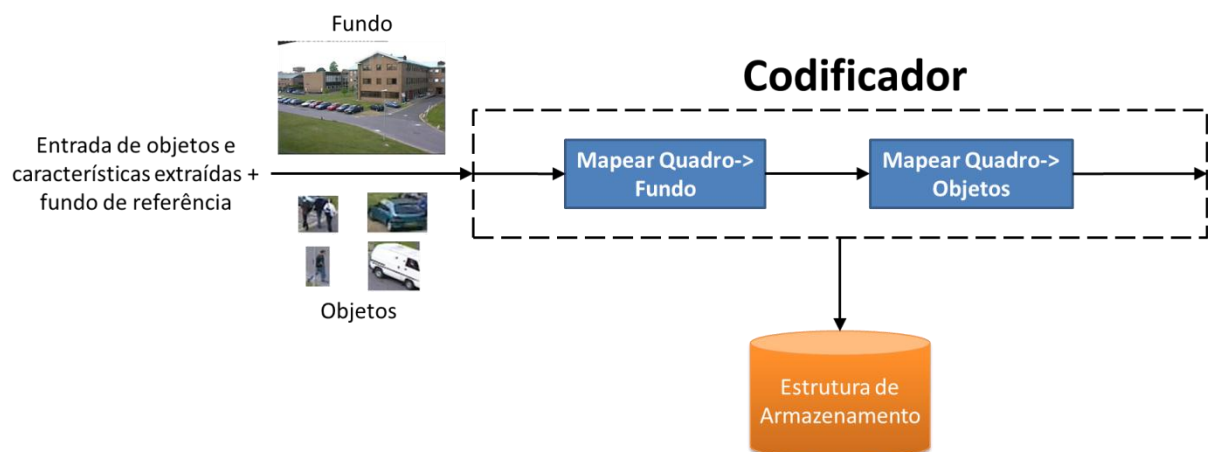


Figura 4 - 3. Diagrama de blocos do codificador.

O codificador realiza um mapeamento do quadro de entrada com uma imagem de fundo (fundo utilizado na etapa de segmentação seção 2.2), em seguida é realizado um mapeamento do quadro com os objetos de entrada e as posições destes na cena. Finalmente, os dados mapeados são guardados numa estrutura de armazenamento. O modelo conceitual de dados dessa estrutura é apresentado na **Figura 4 - 4**.

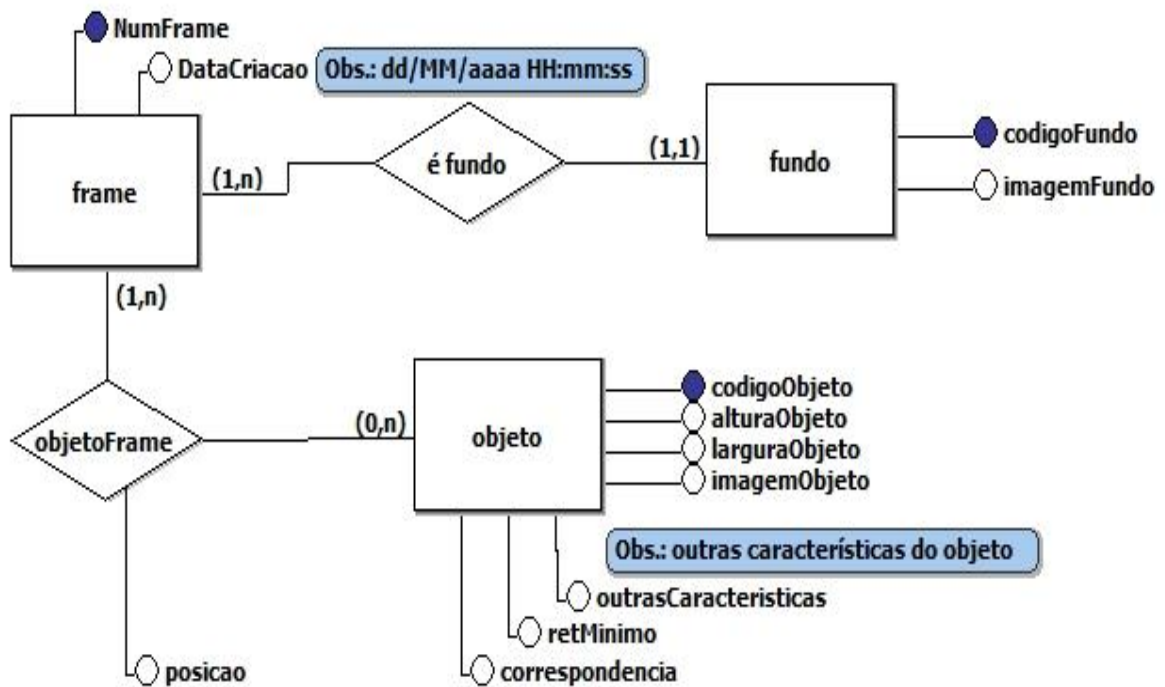


Figura 4 - 4. Modelo conceitual da estrutura de armazenamento.

Este modelo foi concebido de forma a dar total suporte a adaptações na técnica de compressão utilizada. A entidade *frame* representa os quadros de entrada do sistema, cada quadro possui uma numeração auto incrementada e sua data de criação. A entidade *fundo* representa a imagem de fundo de referência para o quadro de entrada, o fundo possui um código e uma imagem. A entidade *objeto* representa os objetos de entrada segmentados com as características extraídas. Alguns dos principais atributos da entidade *objeto* são apresentados na **Figura 4 - 5**, são eles: uma imagem que representa o objeto segmentado, o estado deste objeto (ver seção 3.3) e a correspondência (ver seção 3.1).

Além disso, existem algumas relações entre esses elementos, onde se podem extrair as seguintes informações. Cada quadro está relacionado a uma e somente uma imagem de fundo. Por outro lado, uma imagem de fundo está relacionada a 1 (um) ou mais quadros, ou seja, uma determinada imagem pode ser fundo de vários quadros. Tem-se ainda a relação objetos quadros, onde um quadro pode estar relacionado a zero ou muitos objetos e um objeto deve estar relacionado a um ou vários quadros, onde se deve guardar a posição deste objeto em relação ao quadro.

O mapeamento desse modelo conceitual num modelo lógico é exibido na **Figura 4 - 5**.

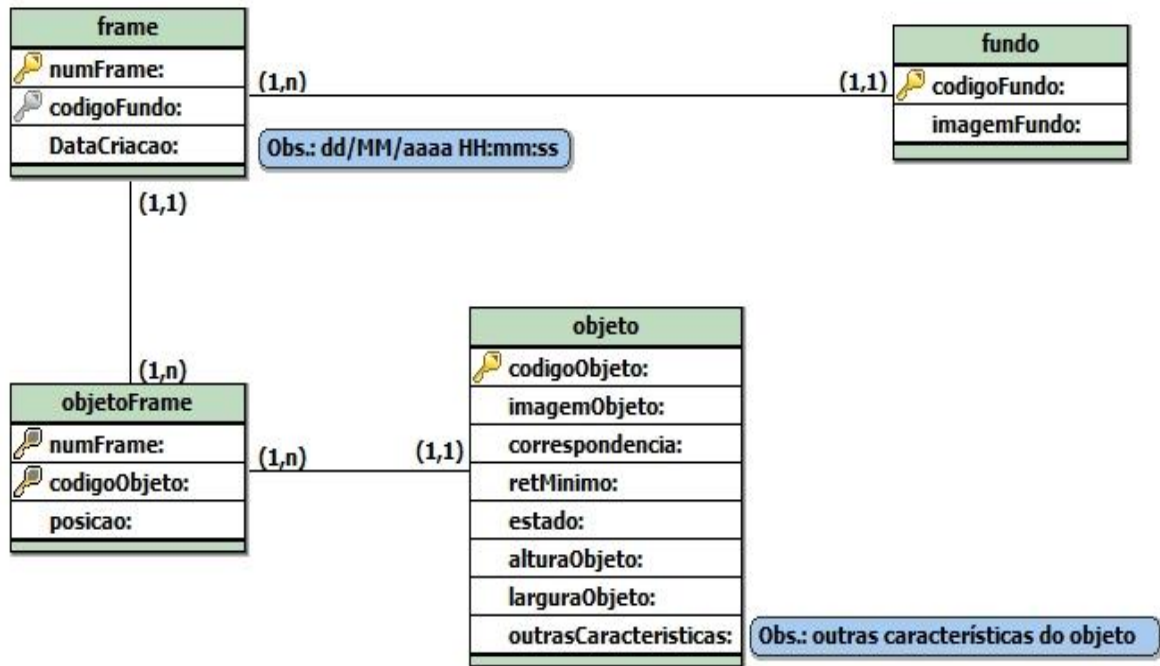


Figura 4 - 5. Modelo lógico da estrutura de armazenamento.

O poder de compressão do sistema proposto é alcançado devido à possibilidade de ter uma mesma imagem de fundo para vários quadros de entrada. Ou seja, toda informação do fundo do quadro de entrada é eliminado, sendo armazenadas apenas as informações de interesse para o sistema de rastreamento. Além disso, é permitido que um mesmo objeto seja referenciado em quadros de entrada diferentes, reduzindo desta forma, o espaço de armazenamento necessário para representar esses objetos. Vale salientar que quando não existem objetos em movimento na cena, nenhuma imagem é armazenada. Necessita-se apenas estabelecer uma relação entre o quadro de entrada e um fundo de referência, que por sua vez não gera carga de armazenamento.

A **Figura 4 - 6** exemplifica um cenário onde uma mesma imagem de fundo é utilizada para vários quadros.

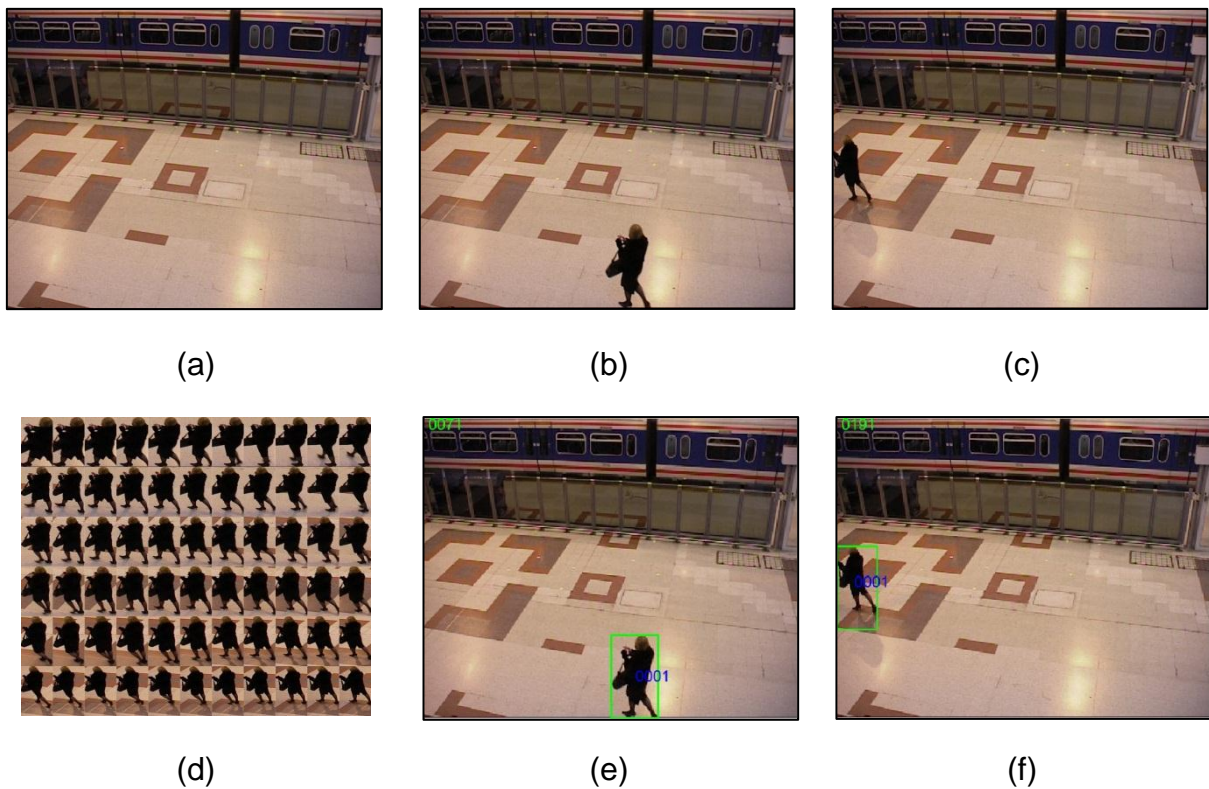


Figura 4 - 6. Cenário com fundo constante.

Pode-se observar que o fundo (**Figura 4 - 6 (a)**) para todas as cenas capturadas (**Figura 4 - 6 (b) (c)**) permanecem o mesmo, desta forma é necessário armazenar apenas as informações dos objetos segmentados (**Figura 4 - 6 (d)**). A **Figura 4 - 6 (e)** e **Figura 4 - 6 (f)** mostram a saída do módulo de rastreamento de objetos.

Na **Figura 4 - 7** é exemplificada a utilização de um mesmo objeto em quadros diferentes. O fundo (**Figura 4 - 7 (a)**) para todas as cenas capturadas (**Figura 4 - 7 (b) (c)**) permanece o mesmo, desta forma é necessário armazenar apenas as informações dos objetos segmentados (**Figura 4 - 7 (d)**), como a abordagem agora permite que um mesmo objeto seja relacionado em mais de uma cena e o objeto que está em movimento é o mesmo, apenas uma imagem é necessário para representá-lo. A **Figura 4 - 7 (e)** e **Figura 4 - 7 (f)** mostram a saída do módulo de rastreamento de objetos.

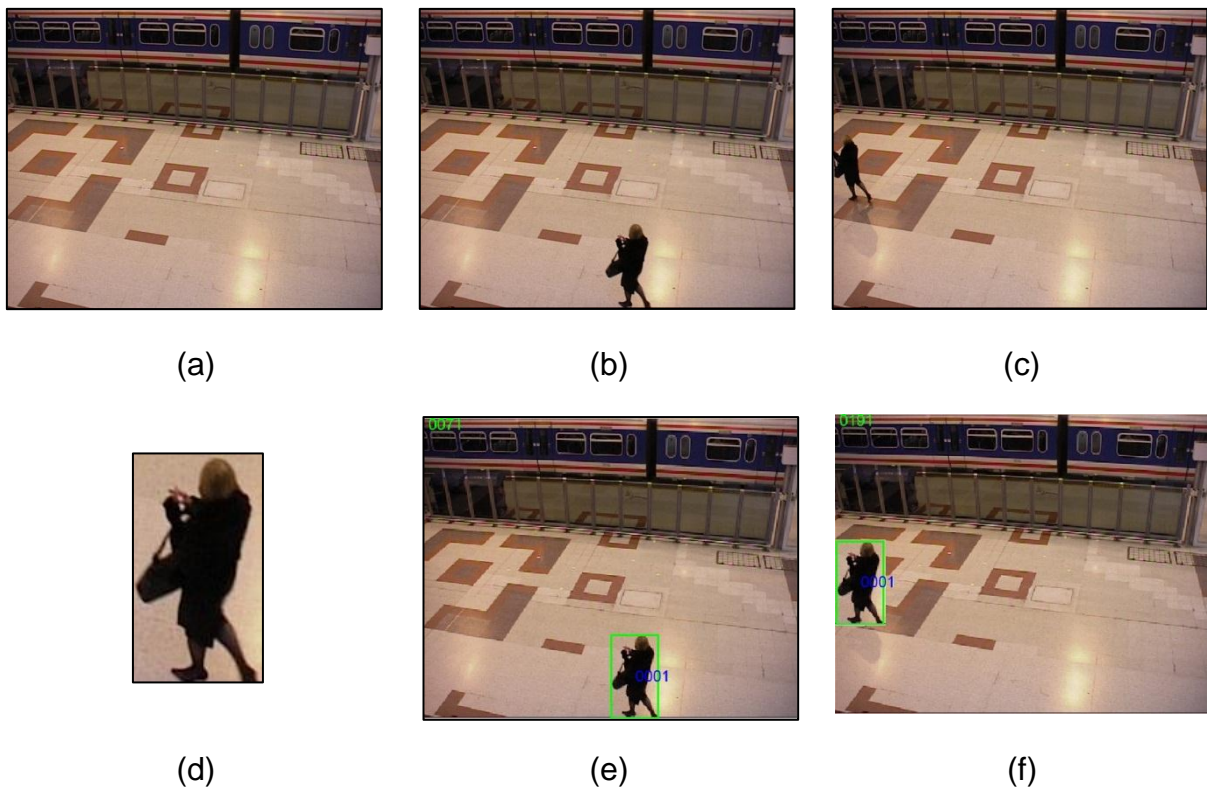


Figura 4 - 7. Cenário com fundo constante e um mesmo objeto por vários quadros.

Na abordagem acima, foram armazenadas apenas duas imagens para representar um conjunto de 120 quadros de entrada. Uma imagem de fundo, e outra imagem do objeto rastreado, variando apenas a posição deste objeto em relação ao quadro. É notório que a quantidade de informação perdida é bem maior que na abordagem anterior. Entretanto, a taxa de compressão é significativamente superior.

A escolha entre qual abordagem utilizar depende do que se necessita da aplicação. Podendo escolher entre, uma abordagem que atualize a cada quadro de entrada a representação visual dos objetos rastreados, ou uma que não apresente atualização. Pode-se ainda, utilizar uma junção das duas abordagens, onde depois de um determinado número de quadros as características dos objetos são atualizadas.

4.2.2 Decodificador

O decodificador realiza o processamento em ordem inversa da mostrada pelo codificador. A descompressão das imagens é iniciada quando as imagens compactadas são passadas para o decodificador. Existem basicamente dois

componentes. O mapeador inverso que realiza o mapeamento em ordem inversa do codificador e o relacionador de objetos, responsável por relacionar objetos, quadros e fundo em uma única imagem. Após esta etapa de processamento a imagem encontra-se pronta para exibição. Como é mostrado na **Figura 4 - 8**.

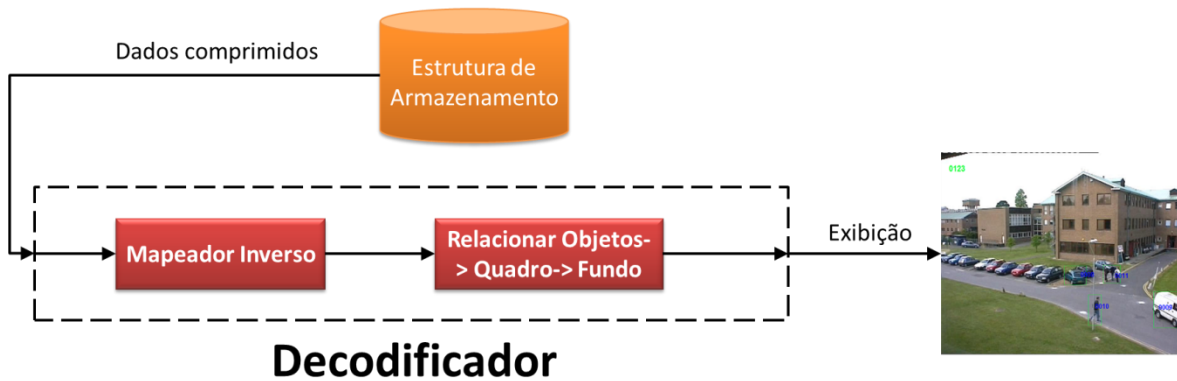


Figura 4 - 8. Diagrama de blocos do decodificador.

A **Figura 4 - 9** exemplifica o processo de decodificação de imagens comprimidas.



(a)



(b)



(c)

Figura 4 - 9. Exemplo de decodificação de imagens.

Para decodificação, é recuperada a imagem de fundo de referência (**Figura 4 - 9** (a)) para o quadro corrente e em seguida, todas as imagens dos objetos em movimentos relacionados a esta cena (**Figura 4 - 9** (b)). Utilizam-se as informações de posição destes objetos para realizar suas projeções no fundo de referência. Por fim, as imagens são exibidas (**Figura 4 - 9** (c)).

Capítulo 5 Arranjo Experimental

Este capítulo discorre sobre o sistema desenvolvido neste trabalho para compressão e indexação de vídeos para sistemas de rastreamento de objetos.

Inicialmente, a seção 5.1 apresenta as bases de dados utilizadas para os experimentos e análises dos resultados do método de compressão CIVRO. Em seguida, na seção 5.2 é discutido sobre a arquitetura do sistema proposto e a saída de cada um dos módulos desenvolvidos. Alguns diagramas de atividades foram criados para melhor exemplificar o fluxo das informações por cada um dos módulos.

Por fim, será apresentada uma análise das técnicas utilizadas para a construção do sistema de rastreamento de objetos por similaridade de características e mostrado a taxa de compressão obtida na utilização do CIVRO para as bases de imagens que serão apresentadas na próxima seção.

5.1 Bases de imagens

Os experimentos foram realizados nas bases de imagens: PETSDATABASE2000¹, PETSDATABASE2001² e PETSDATABASE2006³. Essas bases foram utilizadas no encontro PETS (*Performance Evaluation of Tracking and Surveillance*) [9] que é realizado anualmente desde 2000 com o objetivo de avaliar diferentes técnicas de rastreamento de objetos, tendo como principal foco aplicações voltadas para vigilância e análise de comportamento humano.

As bases escolhidas foram as dos encontros realizados em 2000, 2001 e 2006. Essas bases foram escolhidas de maneira a permitir a validação da técnica proposta para um ambiente misto, com movimento de pessoas e carros, além de permitir análise dos resultados em ambiente externos e interno. A tabela abaixo apresenta um resumo dessas bases.

¹ <ftp://ftp.cs.rdg.ac.uk/pub/PETS2000/>

² <ftp://ftp.cs.rdg.ac.uk/pub/PETS2001/>

³ <ftp://ftp.cs.rdg.ac.uk/pub/PETS2006/>

Tabela 5 - 1. Resumo das bases utilizadas.

Base	Quantidade de Quadros	Tamanho Total (MB)	Resolução	Formato	Tipo de Ambiente	Objetos em Movimentos
PETSDATABASE2000	1452	73,8	(768x576)x3Bytes	JPEG	Externo	Carros e Pessoas
PETSDATABASE2001	2688	152	(768x576)x3Bytes	JPEG	Externo	Carros e Pessoas
PETSDATABASE2006	3021	268	(720x576)x3Bytes	JPEG	Interno	Pessoas

5.2 Arquitetura do sistema

O sistema proposto é composto basicamente de cinco módulos: Captura de Imagens, Segmentação de Movimento, Rastreamento de Objetos, Compressão e Exibição. A **Figura 5 - 1** mostra a visão geral dos módulos desse sistema. Esses módulos foram desenvolvidos com base nos conteúdos apresentados nos capítulos 2, 3 e 4.

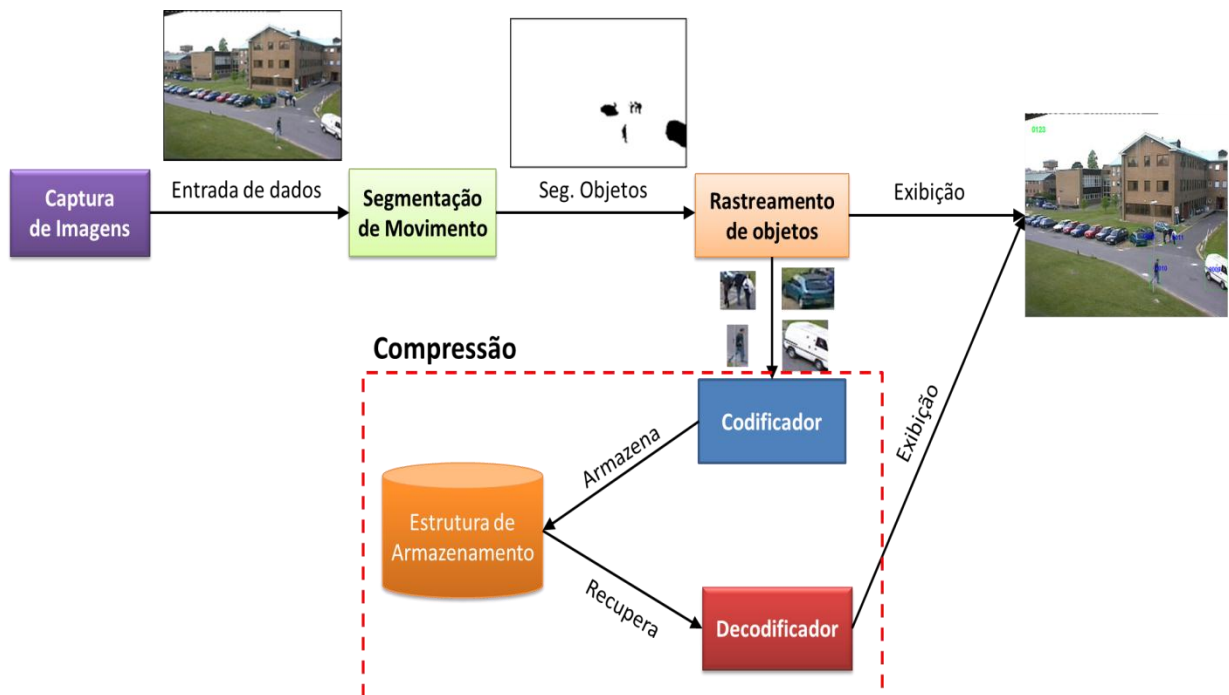


Figura 5 - 1. Modelo geral do sistema.

O processamento dos dados tem início na captura das imagens (módulo de Captura de Imagens), em seguida os objetos em movimento são segmentados

(Segmentação de Movimento) e passados para a etapa de correspondência (Rastreamento de Objetos). Os objetos rastreados são enviados para o módulo de Exibição (que é responsável por realizar as marcações nas imagens) e para o módulo de Compressão (que realiza a codificação e decodificação das imagens). Nas próximas subseções, os módulos acima apresentados serão vistos em mais detalhes.

5.2.1 Captura de Imagens

O módulo de captura é responsável pela alimentação do sistema com os dados de entradas. Esses dados podem ser:

- Imagens sequenciais: conjunto imagens sequencialmente ordenadas, cada imagem corresponde a um quadro de vídeo.
- Vídeo armazenado: arquivo de vídeo obtido do ambiente monitorado e salvo em algum dispositivo de armazenamento.
- Vídeo em tempo real: fluxo de vídeo obtido em tempo real por uma câmera.

Algumas das imagens capturadas das bases de dados utilizadas são mostradas abaixo.

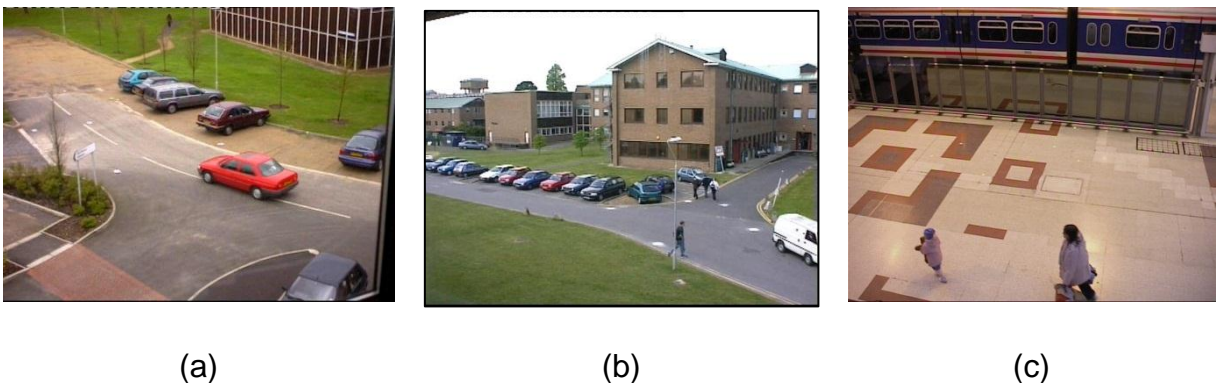


Figura 5 - 2. Exemplos imagens capturas. (a) PETSDATABASE2000, (b) PETSDATABASE2001, (c) PETSDATABASE2006.

O diagrama de atividades deste módulo é ilustrado na **Figura 5 - 3**. Inicialmente é verificado o tipo de dados de entrada. Quando identificado, o primeiro quadro é capturado e passado para o módulo de segmentação de movimento. Em

seguida, é verificado se existem mais imagens a serem capturadas, caso existam, o processo de captura continua. Caso contrário, é terminado.

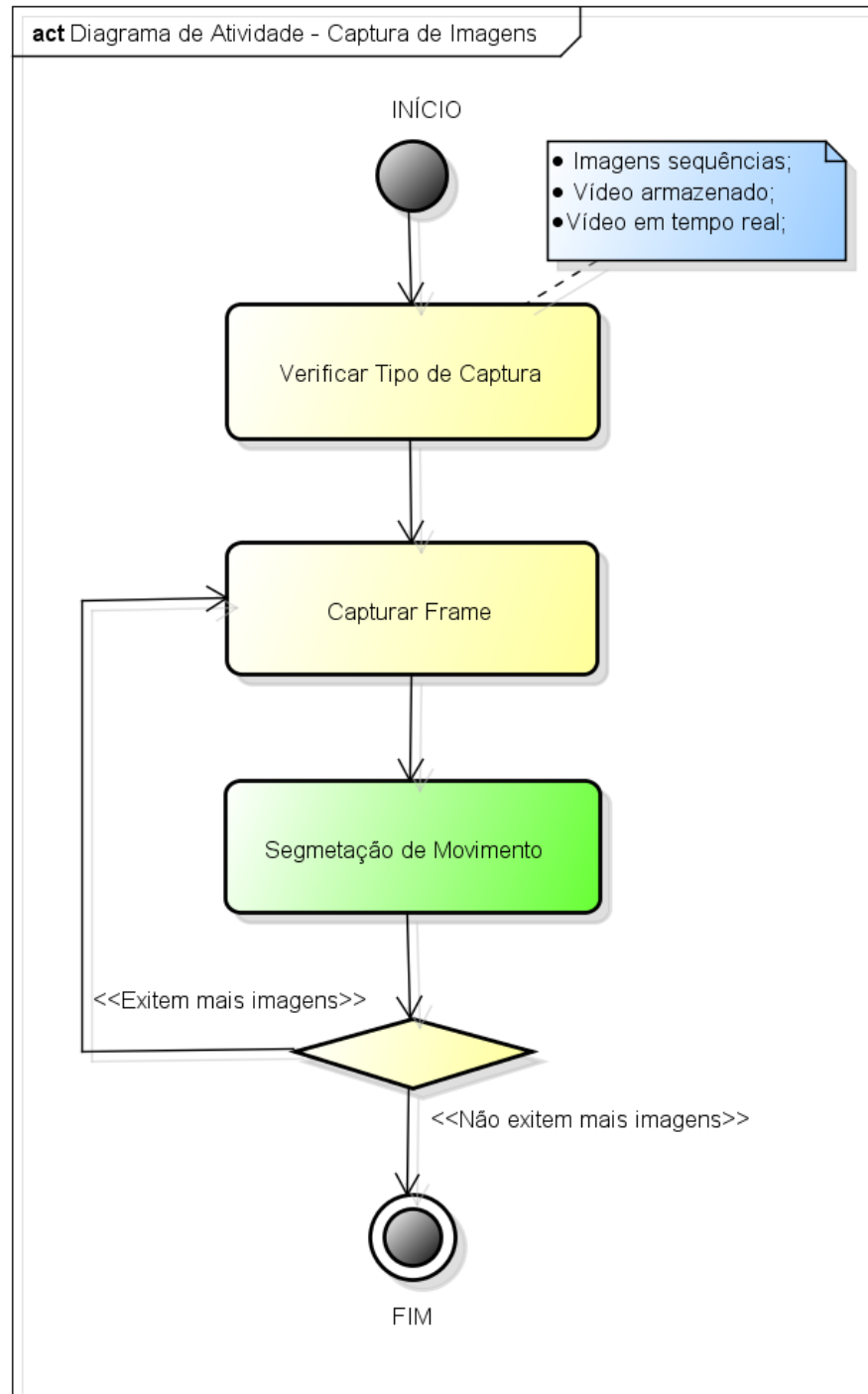


Figura 5 - 3. Diagrama de atividades – Captura de Imagens.

5.2.2 Segmentação de Movimento

O módulo de Segmentação de Movimento é responsável pela separação dos objetos alvos do restante da cena. Este módulo tem como entrada imagens capturadas do ambiente de interesse e possui como saída objetos segmentados que são passadas para o módulo de Rastreamento de Objetos.

A saída do módulo de Segmentação de movimento é mostrada na figura abaixo.

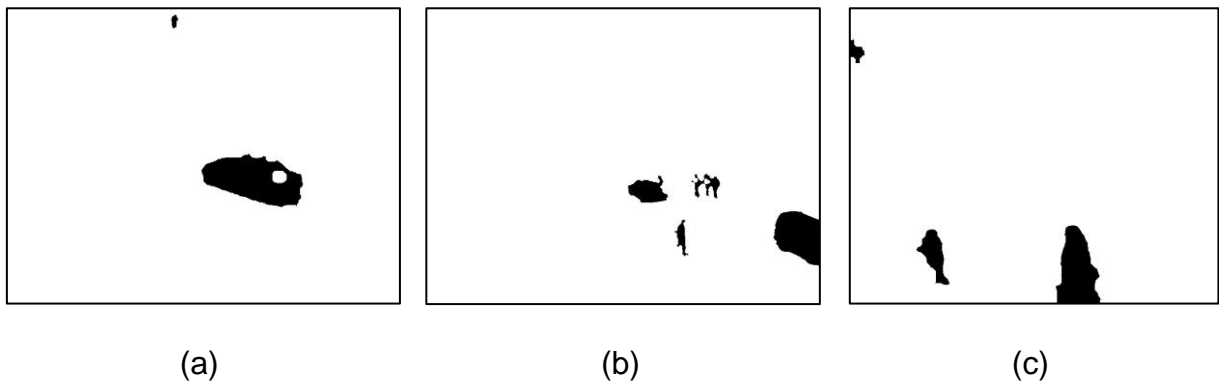


Figura 5 - 4. Saída do módulo de Segmentação de Movimento. (a) PETSDATABASE2000, (b) PETSDATABASE2001, (c) PETSDATABASE2006.

A **Figura 5 - 5** apresenta o diagrama de atividades do módulo de segmentação. Inicialmente é verificado se a imagem de entrada está em escala cinza, se não estiver, a imagem é convertida. Em seguida, realiza-se o cálculo do limiar t utilizado pelo processo de segmentação de objetos (seção 2.2). Por fim, é feita uma etapa de pós-segmentação que visa à diminuição de falhas no processo de segmentação de objetos (como foi descrito na seção 2.3).

5.2.3 Rastreamento de objetos

Após o processamento do módulo de Segmentação de Movimento, os objetos alvos da cena (*foreground*) encontram-se separados dos elementos estáticos (*background*) e estão prontos para a etapa de rastreamento de objetos. O objetivo do módulo Rastreamento de Objetos é estabelecer a correspondência dos objetos alvos detectados no quadro anterior com objetos detectados no quadro atual (Capítulo 3).

Na saída deste módulo, tem-se um conjunto de objetos identificados com suas respectivas características. Essas informações extraídas dos objetos segmentados são essências para os módulos de Compressão e de Exibição. Como foi mostrado no Capítulo 4, a técnica CIVRO utiliza as características extraídas pelo módulo de Rastreamento de Objetos para realizar a indexação dos objetos em rastreamento.

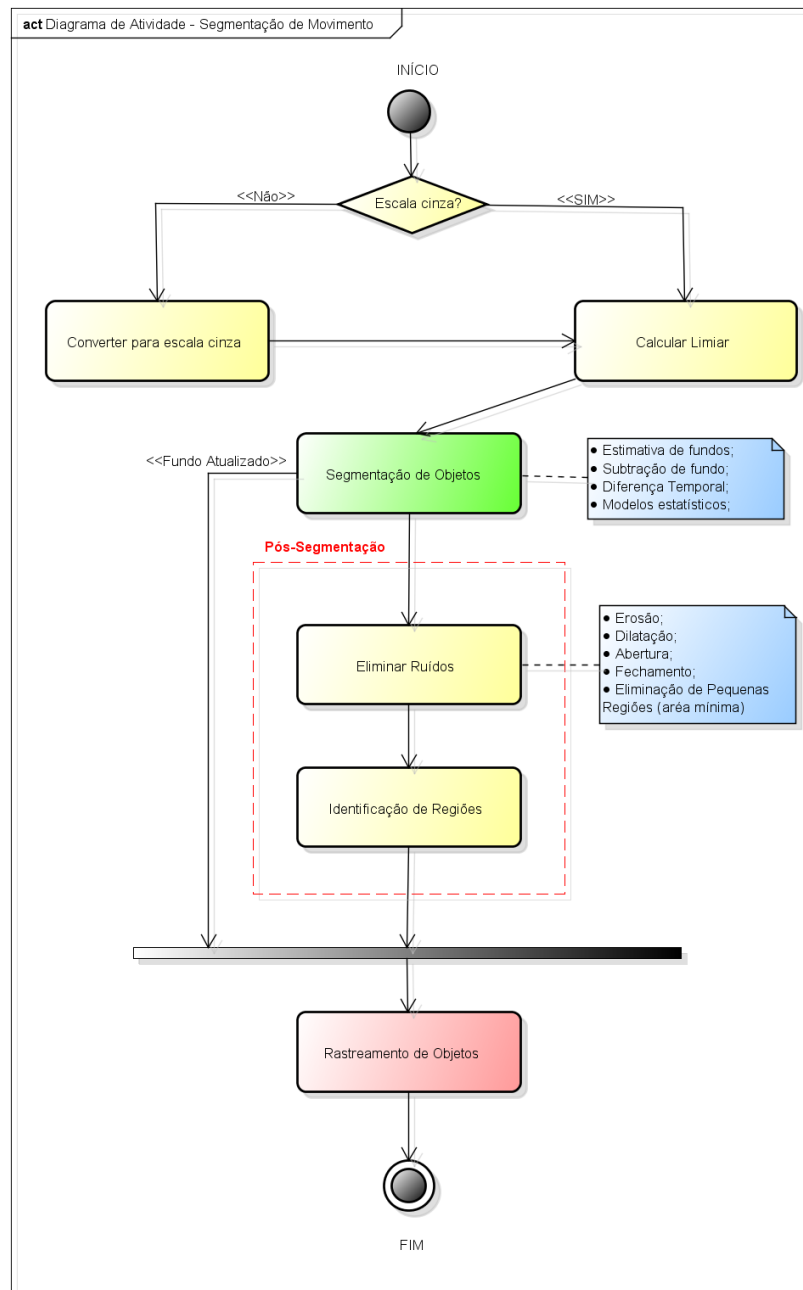


Figura 5 - 5. Diagrama de atividades – Segmentação de Movimento.

A **Figura 5 - 6** apresenta as atividades do módulo de Rastreamento de objetos. Inicialmente é verificado se existem objetos a serem rastreados. Caso positivo, o processamento continua. Os objetos identificados passam por uma etapa de correspondência (**Figura 3 - 1. Fases da etapa de casamento de características [3].**), como foi mostrado na seção 3.1 e em seguida, por uma etapa de detecção e tratamento de oclusão/divisão (ver seções 3.4 e 3.5). Finalmente, os objetos rastreados e suas características extraídas são enviados para os módulos de Exibição e Compressão.

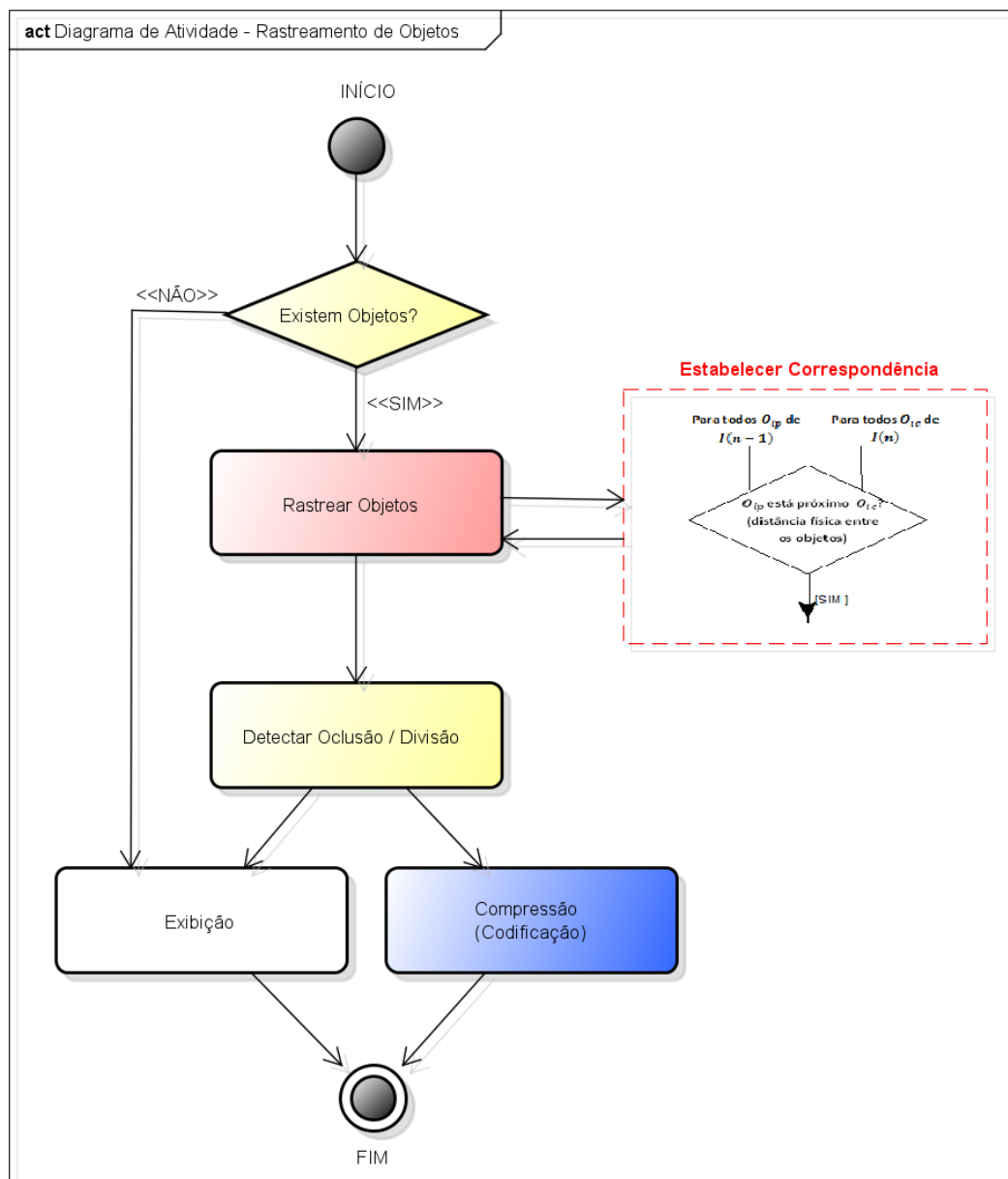


Figura 5 - 6. Diagrama de atividades – Rastreamento de Objetos.

5.2.4 Compressão CIVRO

O módulo de compressão é dividido basicamente em duas grandes etapas. A etapa de codificação (**Figura 5 - 7**), onde os objetos rastreados são recortados da imagem de entrada e se realiza o mapeamento dos objetos segmentados, quadro de entrada e imagem de fundo (como foi descrito na seção 4.2.1). Por fim, esses elementos são armazenados.

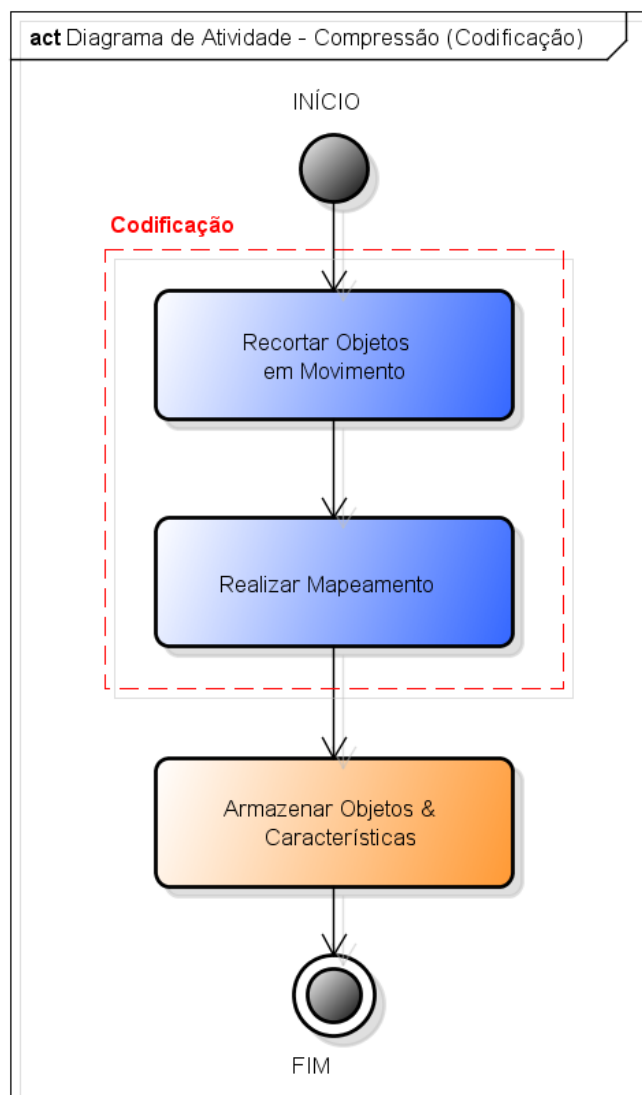


Figura 5 - 7. Diagrama de atividades – Compressão (Codificação).

Como foi discutido no Capítulo 4, apenas as informações dos objetos em rastreamento são armazenados. Sendo assim, o restante da cena é eliminado. A

Figura 5 - 8 exemplifica o conjunto de imagens armazenado para o quadro de entrada corrente.



Figura 5 - 8. Imagens codificadas. (a) PETSDATABASE2000, (b) PETSDATABASE2001, (c) PETSDATABASE2006.

Na etapa de decodificação (**Figura 5 - 9**), realiza-se o processamento inverso do codificador. Inicialmente os objetos são recuperados da estrutura de armazenamento e passam pelo mapeamento inverso (seção 4.2.2). Em seguida é realizado a projeção dos objetos recuperados e o fundo de referência e enviados para o módulo de Exibição.

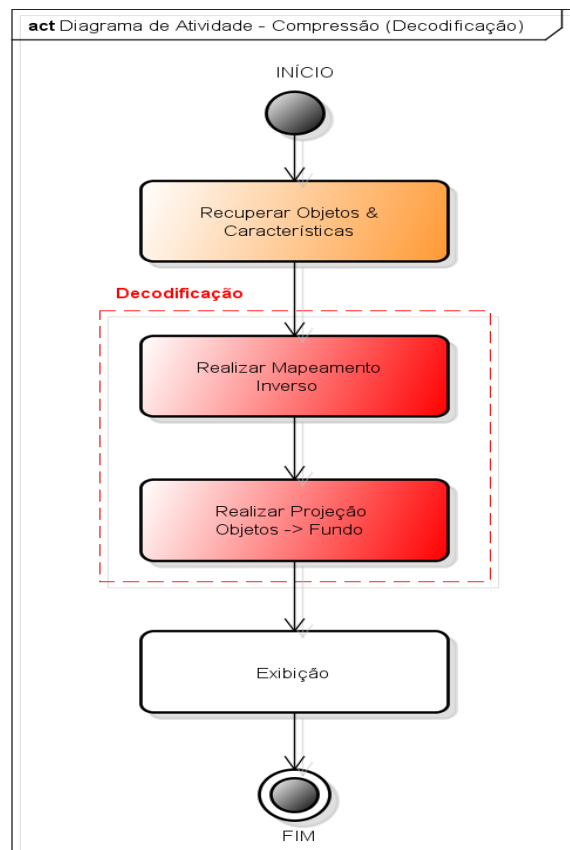


Figura 5 - 9. Diagrama de atividades – Compressão (Decodificação).

Para o processo de decodificação é necessário recuperar os objetos codificados para o quadro de entrada e o fundo de referência mapeado no processo de codificação. A **Figura 5 - 10** apresenta as imagens necessárias para a decodificação.

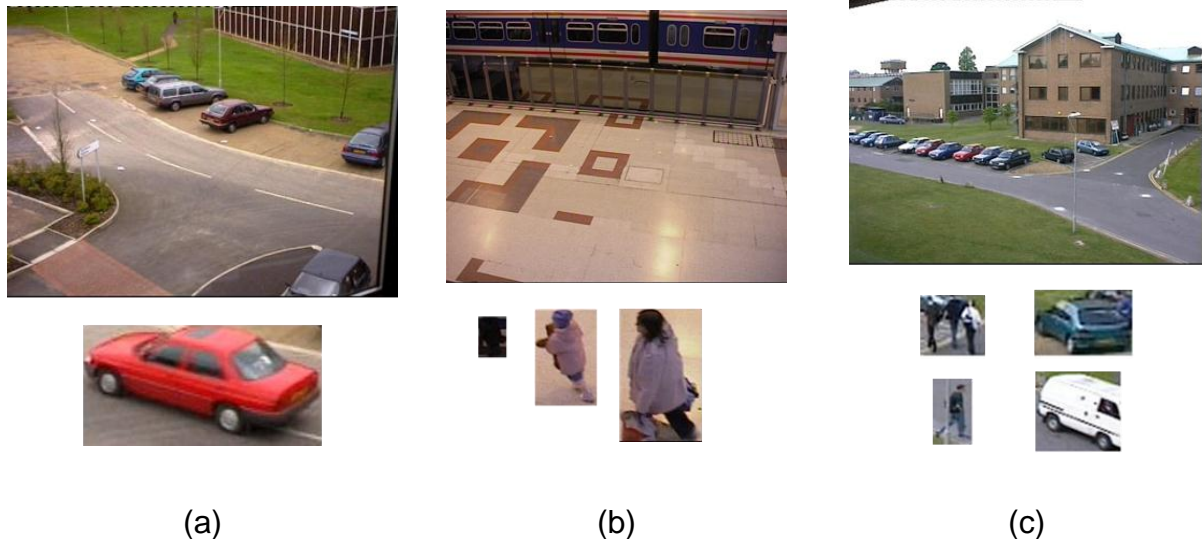


Figura 5 - 10. Imagem de fundo de referência e os objetos codificados para o quadro. (a) PETSDATABASE2000, (b) PETSDATABASE2001, (c) PETSDATABASE2006.

5.2.5 Exibição

O módulo de Exibição (**Figura 5 - 11**) é responsável por exibir as imagens no dispositivo de saída (monitor, por exemplo) e realizar as marcações dos objetos passados pelo módulo de Rastreamento de Objetos ou Compressão.

Inicialmente é verificado se existem objetos de entrada. Caso positivo, inicia-se a etapa de marcação. A etapa de marcação tem como objetivo identificar na imagem de saída às regiões em rastreamento, utilizando informações obtidas pelos objetos segmentados tais como: retângulo mínimo, centro de massa, número do quadro, número do objeto em rastreamento entre outras. Essas informações são apresentadas junto com a imagem de saída (**Figura 5 - 12**).

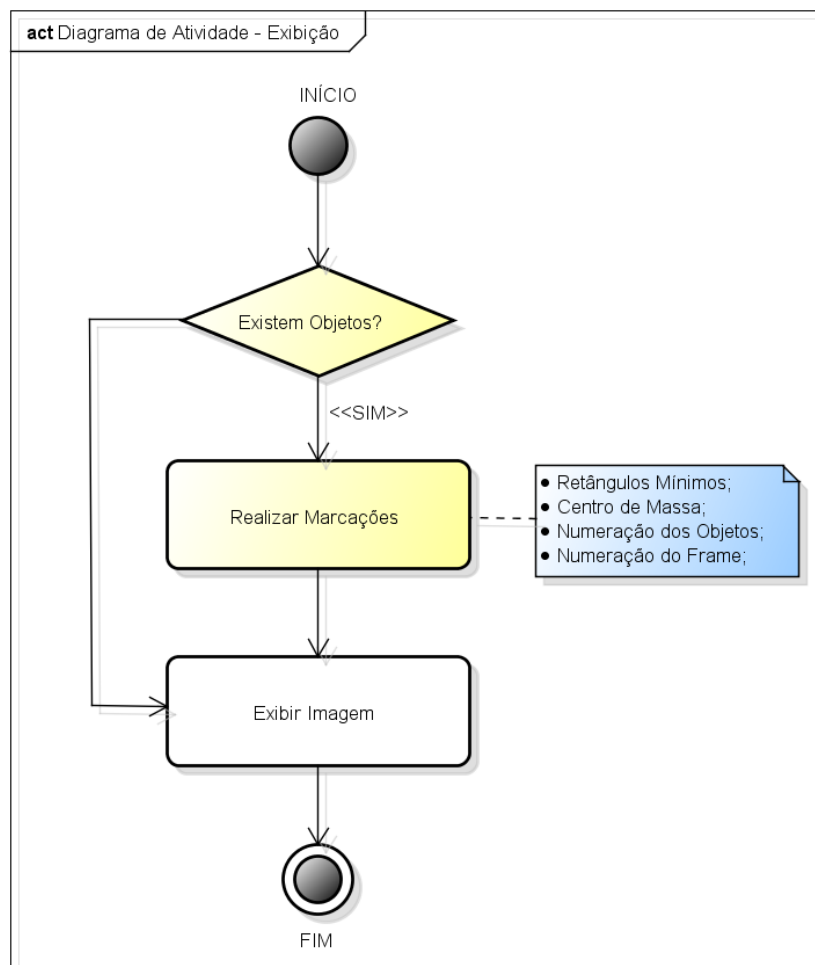


Figura 5 - 11. Diagrama de atividades – Exibição.

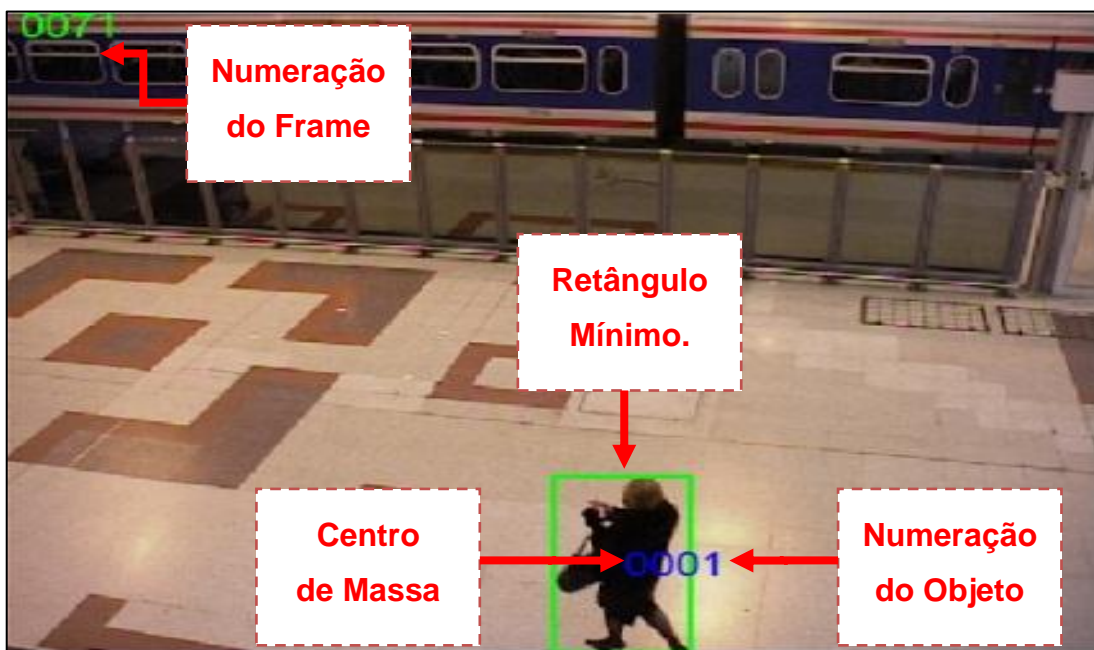


Figura 5 - 12. Saída do módulo de Exibição.

A figura a seguir, apresenta a saída do módulo de Exibição.

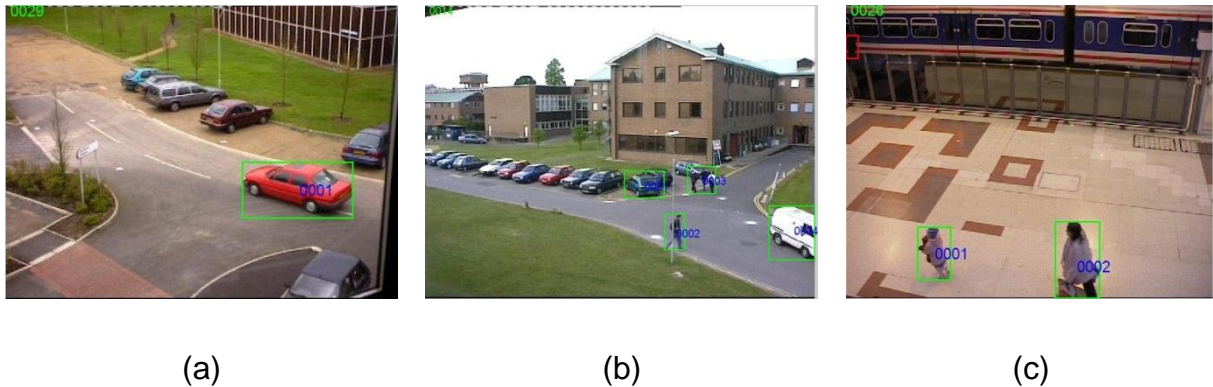


Figura 5 - 13. Saída do módulo de Exibição. (a) PETSDATABASE2000, (b) PETSDATABASE2001, (c) PETSDATABASE2006.

5.3 Análises dos Resultados

Esta seção tem a finalidade de mostrar os resultados obtidos na utilização das técnicas estudadas e da taxa de compressão alcançada na utilização do método CIVRO criado nesta monografia.

5.3.1 Segmentação de movimento

A segmentação de movimento é um dos principais limitadores para o sucesso de uma técnica de rastreamento de objetos, tendo em vista que, se a etapa de segmentação dos objetos em movimento for imprecisa ou ruidosa, serão passados para o módulo de Rastreamento de objetos informações de ruídos da cena causados por alteração de iluminações ou outros fatores, o que irá prejudicar o desempenho do rastreamento gerando falsos positivos.

Como foi mostrado no Capítulo 2, não existe uma técnica de segmentação de movimento que seja adequada para qualquer situação. Porém, pode-se realizar uma etapa de pós-processamento que elimine grande parte dos ruídos e gere uma segmentação um pouco mais limpa.

Na abordagem desenvolvida, foram utilizadas basicamente três técnicas de subtração de objetos em movimento: Subtração de Imagens de Fundos (seção

2.2.1), Diferença Temporal (seção 2.2.2) e Modelo de Mistura de Gaussianas (seção 2.2.3).

Alguns problemas foram encontrados na utilização de cada um dessas técnicas. Quando se utilizou a segmentação por subtração de imagens de fundo alguns resultados indesejados foram obtidos devido à falta de atualização do fundo de referência para o quadro de entrada.

No caso da segmentação utilizando a diferença temporal, a segmentação obtida não tinha uma quantidade de informação suficiente para representar os objetos em movimento, sendo necessário realizar a junção dessas técnicas com outra.

Já a segmentação baseada no modelo de mistura gaussiana apresentou resultados contendo menos ruídos. Entretanto, em algumas situações a atualização do fundo de referência mostrou-se inadequado em alguns casos, cujos objetos em movimento foram considerados como sendo parte do fundo da cena, além do grande custo computacional para realizar esta técnica devido ao fato de utilizar k funções gaussianas para representar cada pixel (como mostrado na seção 2.2.3).

De maneira geral, a técnica que apresentou melhores resultados para o sistema desenvolvido nesta monografia foi a baseada no modelo de subtração de imagens fundo. Entretanto, foi utilizada uma etapa de atualização de fundo de referência após o processamento de um número determinado de quadros de entradas, visando diminuir a quantidade de falsos positivos.

Além disso, a etapa de pós-processamento foi essencial para o desempenho das técnicas de segmentação utilizadas. Destacam-se nesta etapa a erosão, dilatação, abertura, fechamento, preenchimento morfológico de buracos [14] e a utilização do critério de área mínima (onde pequenas regiões são eliminadas).

A **Figura 5 - 14** apresenta um comparativo das técnicas de segmentação de segmentação de movimento utilizadas neste trabalho.

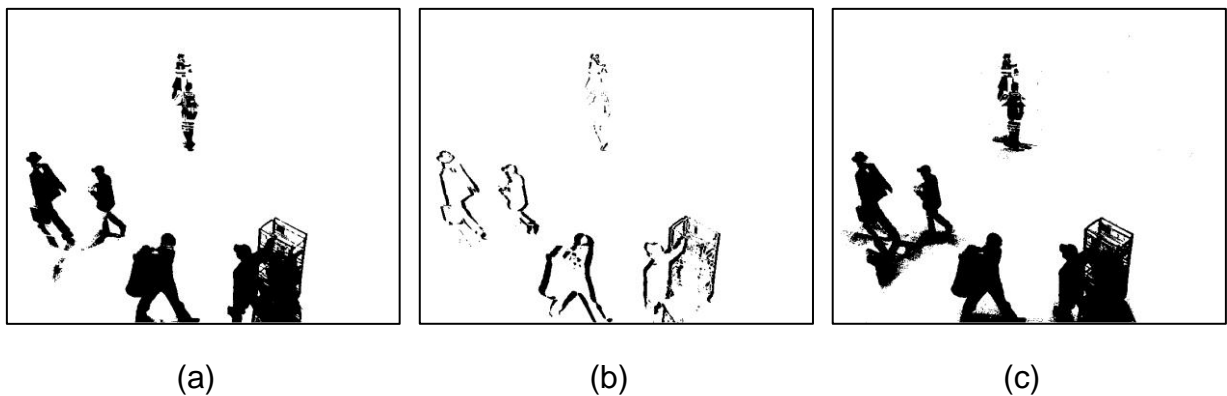


Figura 5 - 14. Comparativo das técnicas de segmentação. (a) Segmentação usando Subtração de Fundo, (b) Segmentação por Diferença Temporal e (c) Segmentação por Modelo de Mistura de Gaussianas.

5.3.2 Rastreamento de Objetos

O rastreamento dos objetos segmentados pelo módulo de segmentação de movimento foi feito utilizando a similaridade de características. Visando diminuir a quantidade de falsos positivos gerada por problemas no processo de segmentação, foram adotados alguns estados dos objetos em rastreamento (como foi mostrado na seção 3.3). Esta abordagem teve uma melhora considerável no resultado final do processo de rastreamento, principalmente em dois estados: no estado "novo" (quando um objeto entra na cena) e no estado "perdido" (quando um objeto não é mais identificado).

Quando um "novo" objeto é identificado ele recebe um rótulo provisório até que seja observado que realmente trata-se de um objeto em movimento e não de ruídos causados pelo segmentador de movimento. Agora, quando se trata de um objeto no estado "perdido", espera-se uma determinada quantidade de quadros para descartá-lo.

O algoritmo de rastreamento desenvolvido foi baseado no trabalho de Amer [3], com algumas adaptações para o contexto geral desta monografia, que permitiu a utilização das informações dos objetos rastreados para a indexação e compressão dos vídeos processados.

As características extraídas de cada objeto rastreado foram apresentadas na **Tabela 3 - 1**. Entretanto, a técnica de rastreamento adaptada neste trabalho permite

que sejam extraídas facilmente outras informações dos objetos em rastreamento o que aumenta consideravelmente sua utilização em diversas aplicações.

Desta forma, a técnica de rastreamento pode ser utilizada em cenários que necessitem de outras características dos objetos, por exemplo: sistema de rastreamento de pessoas e detecção de intrusos, em alguns casos, pode ser necessário realizar um reconhecimento facial das pessoas em movimento por áreas restritas. Sendo assim, seria interessante que para cada pessoa em movimento fossem extraídas informações da face. Para isso, basta utilizar algum descritor de face na etapa de extração de características. Em outro sistema, onde se deseja realizar o monitoramento de tráfego em rodovias, é interessante ter informações sobre a placa dos carros em movimento, cor, velocidade entre outros.

5.3.3 Compressão CIVRO

Como foi mostrada no Capítulo 4, a compressão pelo método CIVRO visa não só a diminuição do espaço em disco necessário para armazenar o conjunto de quadros de um vídeo, mas a realização de um processo de indexação de vídeos utilizando as características extraídas pelo módulo de rastreamento de objetos.

Esta abordagem apresentou-se de extrema importância servindo de motivação para a sua utilização em conjunto com os sistemas de rastreamento e monitoramentos de objetos.

Uma gama de aplicações de rastreamento e monitoramento de objetos podem se beneficiar desta técnica. Principalmente, pelo fato de que junto às imagens comprimidas são mantidas as características dos objetos obtidas pelo módulo de rastreamento. Sendo assim, na recuperação dessas imagens não é necessário realizar novamente o processo de rastreamento de objetos para obter a trajetória realizada por um elemento qualquer da cena. Além disso, outras aplicações podem ser criadas visando à utilização dessas informações para realizar a recuperação de imagens baseado no contexto da cena.

Acrescente-se ainda, o ganho de espaço necessário para armazenar/transmitir os elementos rastreados. A seguir será apresentado um comparativo entre o espaço necessário para armazenar a sequência de imagens das

bases utilizadas para realização dos experimentos e o espaço após o processamento pelo sistema

A compressão é realizada em três cenários.

- No primeiro cenário, as imagens dos objetos rastreados são atualizadas a cada quadro, desta forma, a perda de informação é a mínima possível. Este modelo apresenta o menor nível de compressão;
- No segundo cenário a taxa de atualização das imagens dos objetos é feita de depois de x quadros com x variando entre (5,15 e 30);
- No terceiro cenário uma única imagem do objeto é utilizada para representar os objetos em rastreamento durante toda sua trajetória. Desta forma, apresenta uma grande perda de informações, entretanto a taxa de compressão é máxima. Vale salientar que a perda de informação refere-se apenas às informações visuais, podendo-se manter as características originais dos objetos rastreados, como área dos objetos, histograma e outras características de interesses para o sistema fim.

Para validar os resultados obtidos foram analisadas duas métricas:

1. Taxa de compressão: referente à diminuição de dimensionalidade dos vídeos. É calculada por $tx = \frac{tt}{tc}$, onde tx representa a taxa de compressão, tt é o tamanho total do vídeo sem compressão, e tc é o tamanho final do vídeo com compressão;
2. Economia de espaço em disco: referente à quando de espaço em disco é economizado para armazenar o vídeo comprimido, em relação ao seu tamanho inicial. É calculado por $eed = \left(1 - \frac{tc}{tt}\right) * 100$, onde eed é a economia de espaço em disco, tt é o tamanho total do vídeo sem compressão, e tc é o tamanho final do vídeo com compressão.

Na **Tabela 5 - 2** são apresentados os resultados do processo de compressão para cada uma das bases analisadas. Observando estes resultados, pode-se afirmar que houve um ganho substancial de espaço em disco para armazenar o conjunto de imagens utilizadas neste trabalho. A taxa de compressão variou entre 11,87 a 74,03

e a economia de espaço em disco entre 91,91% a 98,65%. Na realização dos experimentos, notou-se que quanto maior a taxa de compressão menor é a qualidade dos vídeos gerados, tendo em vista que menos imagens são utilizadas para representar o conjunto de quadros original.

A escolha entre qual abordagem de compressão utilizar está diretamente relacionada ao que se deseja rastrear pela aplicação fim. Ou seja, se a aplicação fim se destina em rastrear pessoas em movimento e seus comportamentos, é interessante que qualquer movimento dos objetos monitorados seja capturado, podendo, por exemplo, optar pela técnica apresentada no primeiro cenário. Entretanto, se a aplicação fim está preocupada em rastrear carros em movimento em rodovias, objetos pontuais ou objetos que apresentem uma forma fixa, o segundo ou terceiro cenário é recomendado.

Tabela 5 - 2. Compressão CIVRO.

Base	Atualização em Quadros	Tamanho Total (MB)	Taxa de Compressão	Economia de espaço em disco (%)
PETSDATABASE2000	Original	73,8	0	0
	Cenário 1	5,97	12,36	91,91
	Cenário 2 (5)	2,04	36,18	97,24
	Cenário 2 (15)	1,38	53,48	98,13
	Cenário 2 (30)	1,23	60	98,33
	Cenário 3	1,06	69,62	98,56
PETSDATABASE2001	Original	152	0	0
	Cenário 1	12,8	11,87	91,58
	Cenário 2 (5)	4,58	33,19	96,99
	Cenário 2 (15)	3,23	47,06	97,88
	Cenário 2 (30)	2,9	52,41	98,09
	Cenário 3	2,55	59,61	98,32
PETSDATABASE2006	Original	268	0	0
	Cenário 1	13,2	20,30	95,07
	Cenário 2 (5)	5,5	48,73	97,95
	Cenário 2 (15)	4,24	63,21	98,42
	Cenário 2 (30)	3,96	67,68	98,52
	Cenário 3	3,62	74,03	98,65

Capítulo 6

Conclusão e Trabalhos Futuros

Este trabalho contribui para futuras pesquisas nas áreas de segmentação de movimento, rastreamento de objetos, compressão e indexação de vídeos. Auxiliando o desenvolvimento de novas ferramentas de monitoramento com finalidades acadêmicas e/ou comerciais, como é o caso dos sistemas de vigilância automatizados, sistema de monitoramento de trânsito automotivo, entre outros.

Existem diversas pesquisas relacionadas às técnicas de rastreamento, porém poucas se destinam ao desenvolvimento de uma arquitetura de rastreamento genérica, que possa ser utilizada por outras aplicações com finalidades específicas, como é o caso da proposta de Dedeoglu. A arquitetura da ferramenta desenvolvida nesta monografia foi concebida de forma a permitir a adaptações, podendo assim ser adequada a vários contextos, sendo necessária apenas a modificação das características extraídas pelo módulo de Rastreamento de objetos para o contexto a ser monitorado.

Os problemas de rastreamento de objetos apresentam-se como desafiadores, principalmente devido a fatores relacionados ao processamento de sequência de imagens. Esses problemas são causados devido à introdução do aspecto temporal, tais como deformações dos objetos, oclusões, não uniformidade de iluminação e a manutenção da coerência espaço-temporal ao longo dos quadros.

Além disso, existe uma enorme dificuldade para gerenciar a grande quantidade de informação gerada pelo rastreamento que, de certa forma, limita a ampla utilização destas técnicas. Sendo assim, foi proposta uma abordagem para compressão e indexação de vídeos visando economia do espaço utilizado para armazenamento e agilidade na recuperação de imagens de cenas específicas.

Os resultados obtidos mostraram-se promissores, principalmente a abordagem para compressão e indexação de imagens (CIVRO). De acordo com a análise realizada, obteve-se uma taxa de compressão variando entre 11,87 a 74,03,

apresentando assim, um ganho substancial de espaço em disco para armazenar/transmitir o conjunto de imagens avaliadas neste trabalho.

De maneira geral, pretendeu-se desmitificar várias dificuldades relacionadas ao rastreamento de objetos, além de propor melhorias ao método como as imagens são armazenadas e recuperadas em sistemas de rastreamento e monitoramento.

6.1 Trabalhos Futuros

Como trabalho futuro pode-se desenvolver uma ferramenta com aplicabilidade na área de vigilância e monitoramento de pessoas visando à detecção automática de intrusos e de eventos suspeitos realizados por pessoas no ambiente monitorado; como abandono de uma maleta (possível instalação de uma bomba) ou a remoção de elementos pertencentes ao ambiente monitorado (possível furto).

Outro trabalho seria realizar a junção da técnica de compressão desenvolvida para o contexto de rastreamento de objetos (CIVRO), com técnicas de compressão de propósito geral, como MPEG (*Moving Picture Experts Group*), visando o aumento da taxa de compressão obtida.

Além disso, podem-se extrair outras características dos objetos em monitoramento (tais como expressões faciais, velocidades dos objetos em monitoramento, direção desses objetos dentre outros) visando à melhoria da técnica de rastreamento utilizada e do processo de indexação.

Pode-se ainda, criar um método de recuperação de imagens baseado em conteúdo, que utilize as informações indexadas dos vídeos rastreados para realizar consultas por objetos que possuam determinadas características, ou que satisfaçam a determinados critérios.

A recuperação dessas imagens poderia ser realizada de forma textual, por exemplo, encontrar todas as cenas em que um determinado indivíduo esteja sorrindo, ou de forma gráfica; a partir de uma imagem de entrada, recuperar todas as cenas onde sejam encontrados objetos com características semelhantes, podendo desta forma, ser utilizado para encontrar suspeitos de crimes a partir de uma foto.

Bibliografia

- [1] Yilmaz, A.; Javed, O.; Shah, M. **Object Tracking: A Survey**, ACM Computing Surveys, vol. 38, no. 4, p. 13, Dec. 2006.
- [2] Wang, L.; Hu, W.; Tan, T. **Recent Developments in Human Motion Analysis**, Pattern Recognition, Vol. 36, No. 3, pp. 585-601, 2003.
- [3] Amer, A. **Voting-based Simultaneous Tracking of Multiple Video Objects**, IEEE Trans. on Circuits and Systems for Video Technology, vol. 15, no. 11, pp. 1448-1462, Nov. 2005.
- [4] Dedeoglu, Y. **Moving object detection, tracking and classification for smart video surveillance**, Master's Thesis, Bilkent University, Ankara, 2004.
- [5] Rosales, R.; Sclaroff, S. **Improved tracking of multiple humans with rajectory prediction and occlusion modeling**. In Proc. of IEEE CVPR Workshop on the Interpretation of Visual Motion, Santa Barbara, CA, 1998.
- [6] Welch, G.; Bishop, G. **An introduction to the kalman filter**. Technical report, University of North Carolina, Chapel Hill, NC, USA, 1995.
- [7] Jianguang, L.; Yang, H.; Hu, W.; Tan, T. **Visual vehicle tracking using an improved EKF**, in Asian Conference on Computer Vision, Melbourne, Australia, pp. 296–301, Jan. 2002.
- [8] Costa, Bruno Alexandre Dias da – **Segmentação, rastreamento de objetos e detecção de eventos primitivos com aplicação no monitoramento automático de ações humanas em vídeo**. Campina Grande: 2008.116 f. Dissertação de Mestrado em Ciência da Computação – Universidade Federal de Campina Grande, Campina Grande.
- [9] PETS 2010. Thirteenth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance. **PETS 2010**. Disponível em: <<http://pets2010.net/>>. Acesso em: 03 de Jun. 2011.
- [10] Calic, J.; Campbell, N.; Dasiopoulou, S.; Kompatsiaris, Y. **A Survey on Multimodal Video Representation for Semantic Retrieval**, Computer as a Tool,

2005. EUROCON 2005. The International Conference on, vol.1, pp.135-138, Nov. 2005.
- [11] Veenman, C.; Reinders, M.; Backer, E. **Resolving motion correspondence for densely moving points**. IEEE Trans. Patt. Analy. Mach. Intell. no. 23, 1, 54–72, 2001.
- [12] Siebel, N. T. **Design and Implementation of People Tracking Algorithms for Visual Surveillance Applications**. PhD thesis, Department of Computer Science, The University of Reading, Reading, UK, March 2003.
- [13] Comaniciu, D.; Ramesh, V.; Andmeer, P. **Kernel-based object tracking**. IEEE Trans. Patt. Analy. Mach. Intell. no. 25, pp. 564–575, 2003.
- [14] Gonzalez, C. R.; Woods, R. E. **Processamento Digital de Imagens**. 3. ed. São Paulo : Peason Prentice Hall, 2010.624p.
- [15] Jacques, J. C. S.; Jung, C. R.; Musse, S. R. **A background subtraction model adapted to illumination changes**. In Proceedings of the IEEE International Conference on Image Processing, Atlanta, GA, pp. 1817–1820, 2006.
- [16] Jacques, J. C. S.; Jung, C. R.; Musse, S. R. **Background subtraction and shadow detection in grayscale video sequences**, in Proceedings of SIBGRAPI, Natal, Brazil, pp. 189-196, Oct. 2005.
- [17] Fuentes, L.M.; Velastin, S.A. **Assessment of Image Processing Techniques as a means of Improving Personal Security in Public Transport**, 2nd European Workshop on Advanced Video-based Surveillance, AVBS 2001, Kingston upon Thames, UK, 2001.
- [18] Hall, D.; Nascimento, J.; Ribeiro, P.; Andrade, E.; Moreno, P.; Pesnel, S.; List, T.; Emonet, R.; Fisher, R.B.; Victor, J.S.; Crowley, J.L.. **Comparison of target detection algorithms using adaptive background models**, Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. *2nd Joint IEEE International Workshop on* , vol., no., pp. 113- 120, 15-16 Oct. 2005.
- [19] Collins, R.; Lipton, A.; Kanade, T.; Fujiyoshi, H.; Duggins, D.; Tsin, Y.; Tolliver, D. ; Enomoto, N.; Hasegawa, O. **A system for video surveillance and monitoring: VSAM final report**. Technical Report CMU-RI-TR-00-12, Robotics Institute,

- Carnegie Mellon University, 2000.
- [20] Lara, A. C. **Segmentação de movimento usando morfologia matemática**. São Paulo: 2007.139 f. Dissertação de Mestrado em Ciência da Computação - Universidade de São Paulo, São Paulo.
- [21] Stauffer, C.; Grimson, W. **Adaptive background mixture models for real-time tracking, Computer Vision and Pattern Recognition**. IEEE Computer Society Conference on, vol. 2, pp. 246-252, 1999.
- [22] Kaewtrakulpong, P.; Bowden, R. **An Improved Adaptive Background Mixture Model for Real time Tracking with Shadow Detection**. In Proceedings of European Workshop on Advanced Video Based Surveillance Systems, 2001.
- [23] Lei, Bangjun; Xu, Li-Qun. **Real-time outdoor video surveillance with robust foreground extraction and object tracking via multi-state transition management**, Pattern Recognition Letters, vol. 27, no. 15, Nov. 2006.
- [24] Wang, Y.; Xing, C.; Zhou, L. **Video Semantic Models : Survey and Evaluation**, IJCSNS International Journal of Computer Science and Network Security, Vol. 6 No. 2 pp. 10-20. , Feb. 2006.
- [25] Declair, C.; Hacid, M.-S.; Kouloumdjian, J. **A database approach for modeling and querying video data**, Data Engineering, 1999. Proceedings., 15th International Conference on , pp.6-13, Mar. 1999.
- [26] ALMEIDA, Luís Fernando Barbosa. A Metodologia de Disseminação da Informação Geográfica e os Metadados. Tese de Doutorado. Centro de Ciências Matemáticas e da Natureza – UFRJ. Rio de Janeiro, 1998.
- [27] DAY, M. **Issues and Approaches to Preservation Metadata**. Disponível em: <<http://www.ukoln.ac.uk/metadata/presentations/rlg-npo/warwick.html>>. Acesso em: 03 de Jun. 2011.

Apêndice A

Código Fonte

O código da ferramenta desenvolvida foi disponibilizado em:
<<http://code.google.com/p/civro/source/browse/>>.

Ele é composto por basicamente cinco pacotes:

- Captura_Img: responsável pela alimentação do sistema com as imagens de entradas.
- Seg_Movimento: responsável pela separação dos objetos alvos do restante da cena, realizando a segmentação dos objetos em movimento.
- Rastreamento_Obj: responsável por estabelecer a correspondência dos objetos alvos detectados no quadro anterior com objetos detectados no quadro atual.
- Compressao: é responsável por realizar a codificação e decodificação das imagens, além de realizar a indexação.
- Exibicao: responsável por exibir as imagens no dispositivo de saída, realizando as marcações dos objetos rastreados.