UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE CIÊNCIA DA COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

IDENTIFICAÇÃO DE TRAJETÓRIAS ESPAÇO-TEMPORAIS DE MOVIMENTOS EM VÍDEO

NÚBIA ROSA DA SILVA

Uberlândia - Minas Gerais
2010

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE CIÊNCIA DA COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

NÚBIA ROSA DA SILVA

IDENTIFICAÇÃO DE TRAJETÓRIAS ESPAÇO-TEMPORAIS DE MOVIMENTOS EM VÍDEO

Dissertação de Mestrado apresentada à Faculdade de Ciência da Computação da Universidade Federal de Uberlândia, Minas Gerais, como parte dos requisitos exigidos para obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Banco de Dados.

Orientadora:
Prof^{a}. Dr^{a}. Célia Aparecida Zorzo Barcelos

Uberlândia, Minas Gerais
2010

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE CIÊNCIA DA COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Os abaixo assinados, por meio deste, certificam que leram e recomendam para a Faculdade de Ciência da Computação a aceitação da dissertação intitulada "**Identificação de Trajetórias Espaço-Temporais de Movimentos em Vídeo**" por **Núbia Rosa da Silva** como parte dos requisitos exigidos para a obtenção do título de **Mestre em Ciência da Computação**.

Uberlândia, 28 de Maio de 2010

Orientadora: _____

Prof^a. Dr^a. Célia Aparecida Zorzo Barcelos
Universidade Federal de Uberlândia

Banca Examinadora:

_____

Prof. Dr. Eraldo Ribeiro
Florida Institute of Technology

_____

Prof. Dr. Odemir Martinez Bruno
Universidade de São Paulo

_____

Prof. Dr. Ricardo José Ferrari
Universidade Federal de Uberlândia

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE CIÊNCIA DA COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Data: Maio de 2010

Autor:	**Núbia Rosa da Silva**
Título:	**Identificação de Trajetórias Espaço-Temporais de Movimentos em Vídeo**
Faculdade:	**Faculdade de Ciência da Computação**
Grau:	**Mestrado**

_____

Autor

Dedicatória

Aos meus pais Alcides e Naíma.

Agradecimentos

E é tão bom agradecer. Agradecer cada gesto de amizade, carinho e ajuda daqueles que muitas vezes nem te conhecem e ajudam simplesmente porque ajudar faz bem. Durante a caminhada do mestrado encontrei muitas pessoas que me ajudaram, até mesmo sem saber que o estavam fazendo. Nessas conversas de ônibus, de filas em estabelecimentos e tantos outros lugares, pessoas me motivaram pela sua experiência de vida e por seu carisma. Tenho certeza de que essas pessoas me ajudaram bastante, porque por um momento de suas vidas demonstraram simpatia e sinceridade para comigo. Agradeço a Deus porque encontrei muitas pessoas assim.

Agradeço a Deus principalmente por aquelas pessoas que estão bem mais presentes em minha vida. Deus colocou Alcides e Naíma, meus pais, para serem meu alicerce, me ensinando a ter respeito pelo próximo, caráter e muito mais. Eles me instruíram e por toda a minha vida dedicaram seu tempo a mim, choraram o meu choro e às vezes até meus sorrisos.

Agradeço pela professora Célia por seu profissionalismo, apoio, paciência, orientação e principalmente por sua amizade e cuidado comigo durante todo o tempo. Por causa dela vivi experiências surpreendentes e sou imensamente grata pela confiança que tem em mim.

Pelo professor Eraldo Ribeiro, por sua receptividade quando visitei o Instituto de Tecnologia da Flórida, por sua colaboração para a realização deste trabalho e principalmente por sua amizade.

Pelo professor Marcos Aurélio pela amizade, carinho, preocupação, colaboração e incentivo de crescimento.

Pelos professores da Pós-Graduação em Ciência da Computação da Universidade Federal de Uberlândia, sempre dispostos a incentivar a pesquisa e o desenvolvimento dos alunos. À ex-secretária da Pós-Graduação, Maria Helena, e ao atual secretário, Erisvaldo, pela dedicação, carinho e atenção ao atender os alunos e muitas vezes ouvir os nossos desabafos.

Por todos os meus amigos, tanto da pós-graduação quanto de outros vínculos, porque são o motivo de eu não me sentir sozinha neste mundo, por eu saber que pessoas boas existem e que muitas delas estão perto de mim.

Enfim, agradeço por todos os que contribuíram para a realização deste trabalho, a todos que incentivaram e acreditaram que ele se concretizaria.

"Bem-aventurado o homem que acha sabedoria e o homem que adquire conhecimento."
(Provérbios 3:13)

Resumo

O objetivo deste trabalho é identificar trajetórias de objetos em movimento em vídeos. As trajetórias podem ser usadas para representar o movimento. Ao contrário de estudos com representações baseadas somente na forma, esta proposta utiliza informações de dentro e fora do objeto, não se limitando ao contorno do mesmo, para representação do movimento. O primeiro passo é segmentar cada frame do vídeo em um grande conjunto de regiões denominadas *superpixels*, que mantém características semelhantes de textura e brilho em cada região. Então, cada *superpixel* é caracterizado de acordo com um modelo de aparência e é representado por seu centróide. Estes *superpixels* são monitorados pelos frames do vídeo para gerar as trajetórias. O monitoramento é realizado utilizando uma função que calcula o custo de correspondência das regiões em frames subsequentes, levando em consideração a correspondência entre as regiões e a consistência de suas vizinhanças. As idéias relatadas neste trabalho podem ser aplicadas à tarefa de criar padrões de movimento, que são importantes para identificação de movimentos complexos de objetos e predizer comportamento. Experimentos são realizados em vídeos sintéticos e reais.

Palavras chave: fluxo de movimento; trajetórias; *superpixel*; monitoramento; reconhecimento de movimento

Abstract

The goal of this work is to identify flow lines of moving objects in videos. These flows can then be used to represent the underlying movement. Unlike studies using shape-based representations, our work uses the image information inside and outside the object to build denser flows for motion representation. We begin by segmenting each video frame into a large set of non-overlapping uniform-color and texture regions called superpixels. Then, each superpixel is characterized as a model appearance and is represented by its centroid. These superpixels are then tracked over a number of frames to generate flow lines. Tracking is done using a function that calculates the cost of matching corresponding regions in subsequent frames, taking into account both the correspondence between regions and the consistency of their neighborhoods. The ideas reported in this work can be applied to the task of creating motion patterns, which in turn are important for identifying complex movements of objects and predict behavior. Experiments are performed on a set of synthetic and real-world motion sequences.

Keywords: motion flow; trajectories; superpixel; tracking; motion recognition

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Técnicas utilizadas em imagens também podem ser aplicadas em vídeos. Uma abordagem frequentemente utilizada é tratar o vídeo como sendo somente uma coleção de imagens, ou seja, extrair os quadros do vídeo sendo cada um deles uma imagem sem considerar a relação entre os quadros do vídeo. Muitos sistemas de gerenciamento de informação em vídeo seguem um conceito similar: o vídeo de entrada é analisado para obter imagens individuais. Entretanto, um vídeo não é simplesmente uma coleção de imagens, ele é uma evolução de relacionamentos espaço-temporais e nesta abordagem comumente utilizada, a natureza temporal do vídeo é negligenciada. Este trabalho tenta capturar as informações úteis e importantes em vídeo, que é o movimento de objetos considerando a variação espaço-temporal.

1.1 Motivação

Nos dias atuais diversos são os problemas relacionados à violência, atos de vandalismo, atentados terroristas e roubos. Para tentar solucionar esses problemas, tem-se investido em sistemas de monitoramento em vídeo. No entanto, a quantidade de vídeos gerada é muito grande e a maioria deles jamais fora assistida. Dessa forma, surge a necessidade de analisar as ações que ocorrem nos vídeos automaticamente, de tal forma que a atenção humana seja requerida somente em casos específicos. Analisar os vídeos automaticamente, além de poupar tempo, também reduz o número de falhas, pois ações manuais estão sujeitas a erros.

Descrever automaticamente quais ações estão acontecendo em um vídeo é ainda um problema aberto e importante em visão computacional. Este problema é difícil por diversas razões. Primeiramente, a posição e o movimento da câmera podem interferir na interpretação do movimento, o fundo da cena pode ter movimentos que não interferem nos movimentos principais de um vídeo, por exemplo, o movimento das folhas de uma árvore quando há presença de vento. Segundo, em caso de identificação de movimentos humanos as dificuldades podem estar relacionadas à determinação do número de pessoas

em cada quadro do vídeo e estimativa de onde elas estão, além de identificar o que sua cabeça, seus braços e pernas estão fazendo. Entretanto, encontrar pessoas e localizar seus membros é uma tarefa árdua, porque pessoas se movem rápido, em diferentes velocidades e de forma imprevisível. Também se vestem de forma variada e podem aparecer em diversas poses. Terceiro, descrever o que cada pessoa está fazendo, é um problema não muito entendido, pois não são conhecidas todas as categorias em que se pode classificar as atividades humanas.

Esta dissertação está focada em construir um sistema que identifica as trajetórias do movimento de objetos em vídeos. Uma das aplicações desse trabalho é que: dado um vídeo, seja possível interpretar quais ações estão sendo realizadas no mesmo, fazendo-se a comparação entre trajetórias. Reconhecer movimentos é um componente importante para diversos tipos de aplicações, tais como:

- vigilância automatizada por vídeo - desde a análise de prontuários de pacientes com sinalização automática de comportamento suspeito até em áreas de alta segurança com atividades suspeitas;
- interface homem-computador - tornando possível a idéia de escritórios e casas inteligentes, rastreamento do olhar para dados de entrada em sistemas para computadores;
- monitoramento de tráfego - coleta de informação em tempo real das estatísticas de tráfego para direcionar o fluxo do trânsito.
- indexação e busca de vídeo - anotação automática e recuperação de vídeos em bancos de dados multimídia;
- reconhecimento de gestos;
- análise de eventos esportivos e coreografia de dança; entre outros.

1.2 Abordagem do Trabalho

Neste trabalho a modelagem do monitoramento de ações é realizada utilizando um modelo de representação de regiões em movimento da imagem por meio de um modelo de aparência. Propondo o desenvolvimento de um método para extrair de forma robusta, suave e descritiva o fluxo temporal de movimentos em vídeo. Estes fluxos são representados por trajetórias e estarão aptos a descrever a variação temporal de pontos na superfície interior à borda do objeto. Desta maneira, as trajetórias representarão os fluxos espaço-temporais do movimento para a descrição de ações em vídeos.

A proposta do trabalho pode ser dividida em três partes. A primeira consiste em segmentar cada quadro do vídeo em regiões com um agrupamento local de pixels, com

características comuns entre os elementos agrupados. Essas regiões são denominadas *superpixels*. Após a segmentação, para cada *superpixel*, define-se uma relação de vizinhança utilizando triangulação de Delaunay e faz-se a extração de suas características a partir de um modelo de aparência, o que constitui a segunda fase. E por fim, monitora-se o *superpixel* quadro a quadro utilizando um custo de correspondência das regiões correspondentes, também levando em consideração a vizinhança dos *superpixels*.

De posse das características de cada *superpixel* é possível monitorar seu deslocamento através dos quadros. Para tal, são utilizados os centróides de cada superpixel para representá-los. A abordagem utilizada para realizar o monitoramento baseia-se no algoritmo *Iterated Conditional Modes* (ICM). Esse algoritmo tem seus fundamentos em Campos de Markov e, simultaneamente, estima os parâmetros de aparência local do *superpixel* e a consistência da vizinhança, melhorando a representação espaço-temporal do movimento.

O fluxo de trajetórias obtido é composto pelas trajetórias de cada ponto monitorado. Os movimentos não necessitam estar definidos do começo ao fim do vídeo. As trajetórias serão agrupadas de acordo com sua semelhança para definir padrões de movimento. Esses padrões serão utilizados para estabelecer uma nova representação de movimentos de objetos em vídeo. Diferentemente dos trabalhos que utilizam a representação baseada na forma, esse trabalho utiliza todas as informações da imagem para construção de fluxos mais densos para melhor representar o movimento.

1.3 Contribuição

A principal contribuição deste trabalho será o desenvolvimento de um estimador de fluxos espaço-temporais, representados por trajetórias. Para tal é realizado o monitoramento do movimento por meio da identificação de trajetórias espaço-temporais do movimento. O estimador incluirá uma descrição local da informação da aparência da região a ser monitorada e permitirá a detecção de fluxos consistentes nos pontos distribuídos em todo o objeto. Posteriormente, este fluxo do movimento poderá ser utilizado para identificar padrões de movimento.

1.4 Organização da Dissertação

Esta dissertação faz uma revisão da literatura relacionada com o método proposto, descreve este método e mostra a eficiência do mesmo por meio de vários experimentos. Este trabalho encontra-se organizado em 5 capítulos, sendo este o primeiro capítulo.

No Capítulo 2 tem-se as prelimirares da dissertação, onde são discutidos temas teóricos pertinentes ao assunto principal deste trabalho.

No Capítulo 3 tem-se um levantamento das metodologias de monitoramento, que diferem principalmente no tipo de representação do objeto, nas características da imagem a serem usadas, na aplicação do tipo de movimento e no modelo de aparência. As abordagens para monitoramento podem ser divididas em monitoramento baseado em pontos, baseado em contorno e em modelos de aparência.

No Capítulo 4 será descrito o método proposto, ilustrando descritivamente como é realizada cada uma das três etapas do trabalho. Também serão discutidos os pontos relevantes para a realização do mesmo.

No Capítulo 5 serão mostrados diversos experimentos realizados para validação do método, bem como, as premissas para obter bons resultados com o método proposto. Os experimentos são compostos por vídeos reais com movimentos humanos e vídeos sintéticos com movimentos variados.

No Capítulo 6 estão as conclusões relativas à proposta do trabalho e os resultados obtidos. De acordo com os resultados foram geradas várias perspectivas para trabalhos futuros e aplicações, que também serão abordadas neste capítulo.

Capítulo 2

Preliminares

Neste capítulo serão definidos alguns conceitos relacionados aos temas que serão tratados nos próximos capítulos. A Seção 2.1 aborda os dois métodos de segmentação em superpixels utilizados por este trabalho. O primeiro deles é o método proposto inicialmente por Ren e Malik [42]. E o segundo é o algoritmo Turbopixels [25] que também segmenta uma imagem em superpixels. A Seção 2.2 trata sobre os Campos de Markov que serão utilizados para definir o modelo de fluxo do movimento.

2.1 Segmentação em *Superpixels*

Humanos têm uma maneira semântica de interpretar tudo ao seu redor, principalmente imagens. Ao observar uma imagem um humano consegue perceber as regiões e os objetos que compõem a mesma e também o seu significado no contexto da imagem. Por outro lado, é uma tarefa complexa interpretar computacionalmente uma cena a partir de sua imagem, definindo padrões, tais como, padrões de textura e sombreamento.

Em geral, humanos não tem problema para interpretar imagens e podem reconhecer objetos e diferentes regiões da imagem, realizando operações, tais como, agrupamento de regiões facilmente. O agrupamento de regiões de acordo com a percepção humana foi estudado pela Psicologia da Gestalt[1], na qual são evidenciados os fatores que permitem aos seres humanos realizarem uma separação perceptual, agrupando regiões em uma cena, ao invés de objetos componentes.

Alguns dos fatores de Gestalt mais relevantes são similaridade, proximidade, continuidade, simetria, agrupamento, fechamento e familiaridade [51]. De acordo com essa abordagem elementos similares e elementos próximos tendem a se agrupar, contornos contínuos são preferidos àqueles com quebra ou outras combinações, mais complexas. O agrupamento está baseado nas propriedades simétricas, pois elementos simétricos são mais facilmente agrupados que aqueles não simétricos. As formas são compreendidas de acordo

[1]Gestalt é uma palavra alemã que significa “configuração” ou “padrão”.

com o conhecimento prévio, ou seja, certas formas só são compreendidas se já forem conhecidas, ou se há uma consciência prévia da sua existência. Se uma forma inteira já tiver sido vista em um certo momento e num outro momento somente uma parte dessa forma estiver sendo vista, a forma inteira poderá ser reproduzida na memória por conhecimento prévio da mesma.

Condições do todo determinam de que forma uma parte será interpretada, por exemplo, ao observar duas cores em uma imagem, as sensações percebidas são determinadas pelas condições da situação de estímulo inteiro. Dessa forma, a questão é investigar as condições do todo para determinar quais são as influências sobre as partes, pois um padrão local pode dar origem a uma figura unitária homogênea ou uma figura articulada com diferentes partes. Experimentos têm mostrado a veracidade dessa afirmação, mostrando que deve-se avaliar as condições "todo-parte" [2].

Baseado nessa forma de percepção humana de agrupamento, muitos trabalhos de segmentação têm utilizado os princípios da Gestalt para seus algoritmos de agrupamento [35, 37, 41, 28]. A segmentação de imagens pode ser classificada em dois segmentos: segmentação baseada em região e baseada em contorno [13]. A primeira é baseada no particionamento de uma imagem em regiões que são similares de acordo com um conjunto de critérios pré-definidos. Na segunda abordagem, a imagem é particionada de acordo com as mudanças abruptas de intensidade, como por exemplo, uma aresta.

Ren e Malik [42], motivados pelas premissas de que pixels são uma representação discreta de imagens e que o número de pixels é muito grande em boas resoluções, tornando as operações em nível de pixels intratáveis, proporam uma solução de agrupamento de pixels em regiões coerentes, com características comuns de cor, textura e brilho dentro da região.

Foi baseando-se nos princípios de Gestalt que Ren e Malik [42] desenvolveram um classificador para combinar características como contorno, textura, brilho e continuidade, agrupando-as. Essas regiões com características semelhantes são denominadas *superpixels*. Desta maneira, os *superpixels* deveriam formar um bom agrupamento de pixels, resultando em uma boa segmentação da imagem em regiões, de tal forma que fosse possível identificar uma mesma região em frames semelhantes.

2.1.1 *Superpixels*

Existem diversas formas de agrupar as regiões similares em *superpixels*, a mais tradicional é a introduzida por Ren e Malik [42], na qual um *superpixel* é um agrupamento local de pixels, com características comuns entre os elementos agrupados e que preserva uma estrutura necessária para segmentação. Uma maneira de segmentar uma imagem em *superpixels* é utilizar o algoritmo *Normalized Cut* [46, 47, 28]. Esse algoritmo é aplicado à imagem para obtenção do mapa de *superpixels*, que é a imagem totalmente segmen-

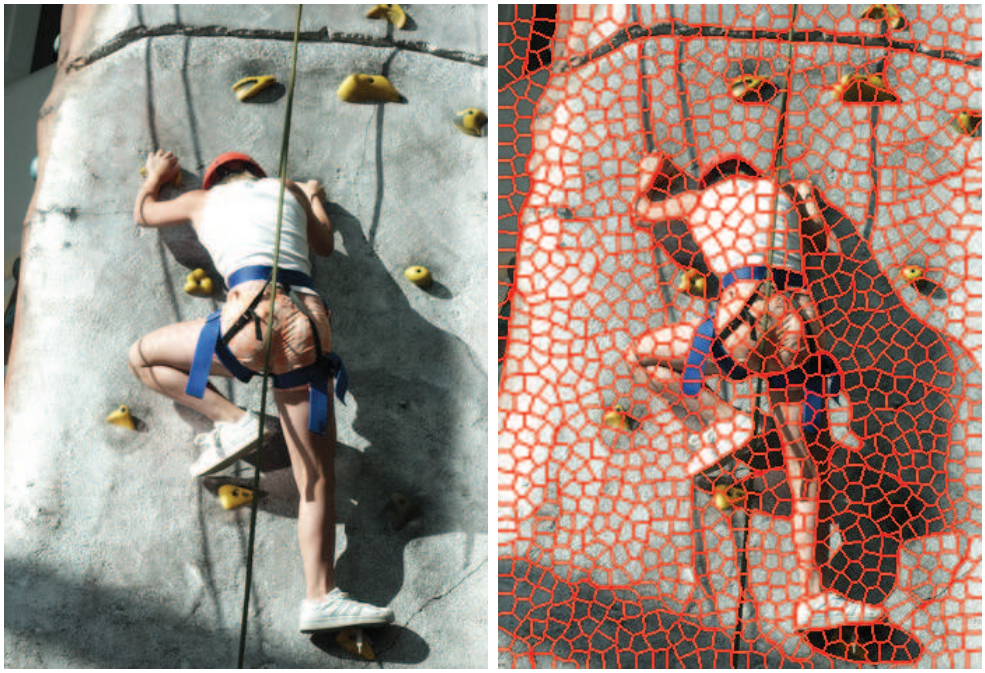
tada em *superpixels*. O *Normalized Cut* particiona a imagem em regiões disjuntas com coerência dos atributos contorno e textura.

Algoritmo *Normalized Cut* trata as características da imagem de forma interligada, formando um grafo ponderado, onde os pixels são os nós do grafo e o peso das arestas ligando os nós é definido por uma função de similaridade entre os pixels, considerando características, tais como, brilho, textura e cor.

No algoritmo *Normalized Cut*, a segmentação de imagens é tratada como um problema de particionamento de grafos, no qual é mensurada a dissimilaridade entre diferentes grupos e a similaridade entre os elementos do mesmo grupo.

A formulação mais popular desse agoritmo, referenciada como *N-Cuts*, é a desenvolvida por [47, 46] e é a base para o algoritmo original dos *superpixels* [42].

Como considerado em [42], *superpixels* são bastante homogêneos em sua forma e tamanho, para simplificação de custos computacionais. A figura 2.1 mostra um exemplo de segmentação com aproximadamente 1000 *superpixels*.



(a) (b)

Figura 2.1: Exemplo de mapa de *superpixels*. (a) Imagem original. (b) Mapa de *superpixels* da imagem (a).

Os *superpixels* devem ter os elementos que o formam similares e os pixels pertencentes a *superpixels* diferentes, dissimilares. Para definir a *similaridade* entre os *superpixels* foram definidos os seguintes conceitos [42]:

Similaridade intra-região: os elementos na região delimitada pelo *superpixel* são definidos como similares de acordo com os parâmetros de brilho, textura e fracos contornos no interior dessa região.

(Dis)similaridade inter-região: os elementos em diferentes *superpixels* são dissimilares, ou seja, eles têm diferentes brilho e textura e alta energia do contorno em pontos de fronteira dos *superpixels*.

A similaridade da textura é analisada utilizando *textons*. O termo *texton* foi primeiramente definido por Julesz [18] como a unidade elementar de percepção da textura e posteriormente re-definido como a combinação de co-ocorrência da saída de filtros lineares orientados [29]. A Figura 2.2 mostra um exemplo de banco de filtros de textura. O processo de modelagem dos *textons* consiste em aprender um dicionário de *textons* contendo modelos de todos os elementos de textura representativos no conjunto de treinamento. Um vetor quantizado da imagem textura é construído baseado no dicionário de *textons*, chamado de "mapa de *texton*" [9]. Aplicando-se o banco de filtros com várias orientações e de acordo com as saídas dos filtros, os pixels são agrupados em canais de *textons*.

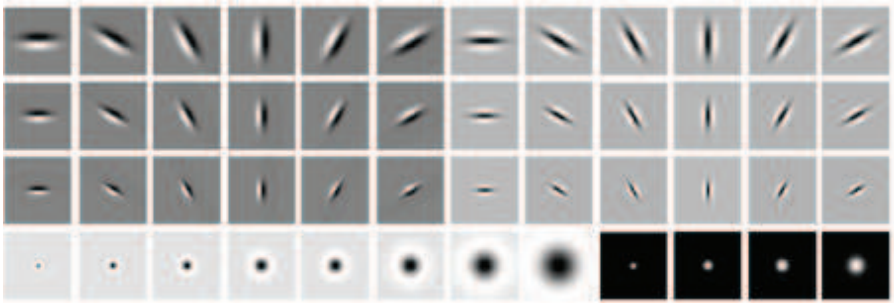


Figura 2.2: Filtros de textura [23].

O estágio de classificação dos pixels em cada *superpixel* é baseado na medida de similaridade χ^2 entre os histogramas de frequência dos mapas de *textons*. A similaridade χ^2 é dada por:

$$\chi^2(p,q) = \sum_i \frac{(p_i - q_i)^2}{p_i + q_i} \tag{2.1}$$

onde p_i e q_i são células de dois histogramas de mapas de *textons*.

A medida da distância χ^2 entre os histogramas dos valores de intensidade de brilho de cada *superpixel* também fornece a similaridade inter-região e intra-região dos *superpixels*. A energia do contorno é computada no nível dos pixels, através do cálculo da energia da orientação em cada pixel. Uma transformada não linear converte a energia da orientação em um contorno suave. A energia do contorno inter-região é o somatório de todos os pixels desse contorno suavizado sobre todos os pixels da fronteira da região que engloba o *superpixel* e a energia do contorno intra-região é o somatório de todos os pixels do contorno suavizado sobre todos os pixels da fronteira dos *superpixels* que se encontram nessa região. Outra característica observada é a boa continuação do contorno, na qual é medida a continuidade curvilinear da fronteira, através da média das mudanças do ângulo tangente de todos os pares de pixels na fronteira que está sendo analisada.

2.1.2 Segmentação em *Superpixels* Utilizando Algoritmo *TurboPixels*

O algoritmo *TurboPixels* introduzido por [25], assim como no trabalho de [42, 32], também segmenta uma imagem em regiões denominadas *superpixels*. Com a segmentação é gerada uma malha de regiões compactas que combina um modelo de evolução de curva para dilatação de pixels espalhados por toda a imagem com um processo de esqueletização da região da imagem que ainda não foi atingida pela dilatação dos pixels. Esse processo de esqueletização do fundo da imagem ocorre para prevenir o encontro das regiões sendo expandidas ao redor do ponto, ou seja, para prevenir a fusão de sementes (pixels iniciais que serão dilatados).

A segmentação, além de considerar os conceitos de similaridade de cor, brilho e textura, baseia-se nos princípios de uniformidade, conectividade, compacidade, suavidade e não sobreposição de *superpixels*, definidos a seguir:

Uniformidade de tamanho e cobertura São escolhidas k sementes (coordenadas que identificam pixels) em toda a imagem. Essas sementes serão expandidas por meio de fluxo geométrico até cobrirem totalmente a imagem. A região de expansão da semente será o superpixel, que por sua vez possui forma e tamanho uniformes, minimizando a região de segmentação de tal forma que o tamanho de cada *superpixel* é comparável com o tamanho da menor região possível de ser segmentada de acordo com a quantidade de sementes definidas.

Conectividade O *superpixel* é formado por um conjunto conecto de pixels, o que é garantido pela dilatação das sementes.

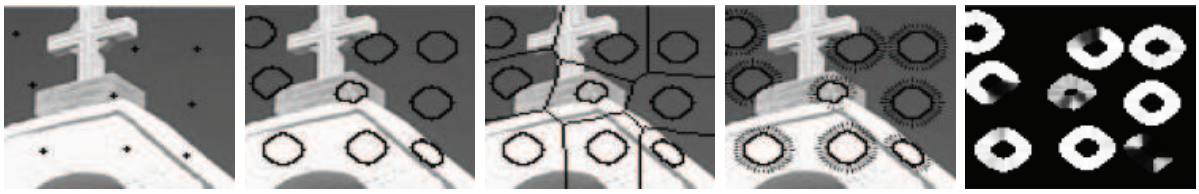
Compacidade Se não houver informação de arestas, os *superpixels* devem permanecer compactos, ou seja, fechados e limitados. Para aumentar a compacidade, foi incluído um termo que produz movimento extrínseco constante nas regiões de intensidade normal.

Suavidade Não deve haver descontinuidade, ou seja, com a dilatação, os limites do *superpixel* deverão coincidir com as arestas da imagem.

Não sobreposição de *superpixels* Cada pixel da imagem deve pertencer a somente um *superpixel*. O crescimento do *superpixel* deve parar quando duas sementes distintas, devido a sua expansão, estiverem próximas de colidir. A colisão deve ser prevenida, pois todas as sementes seriam expandidas até preencherem toda a imagem, o que não resultaria em uma segmentação.

A Figura 2.3 mostra os passos do processo de segmentação em *TurboPixels*, que pode ser sintetizado da seguinte maneira: primeiramente são escolhidas k sementes a serem expandidas em *superpixels*. O número de sementes é inversamente proporcional ao tamanho

das regiões, ou seja, quanto maior for o valor de k menores serão as regiões. As k sementes são dispostas em forma de uma malha na imagem, tal que a distância Euclidiana entre as sementes é $\sqrt{\frac{N}{k}}$, onde N é o número de pixels da imagem.



(a) (b) (c) (d) (e)

Figura 2.3: Passos do algoritmo *TurboPixels* [25]. (a) Distribuição das k sementes na imagem; (b) Evolução das sementes; (c) Atualização do esqueleto do fundo da imagem; (d) Atualização das velocidades; (e) Extensão das velocidades.

Os próximos passos, Figuras 2.3 (b)-(e), são realizados iterativamente até que não haja uma evolução possível, ou seja, até que não haja mais regiões para que as sementes possam ser exapandidas. O primeiro passo dessa iteração, e segundo passo do algoritmo, é realizar a expansão das sementes até que qualquer ponto na fronteira da região que está sendo expandida atinja os limites possíveis, isto é, até que não tenham mais pixels a serem atingidos. O parâmetro chave para controlar essa evolução é o produto de duas velocidades. A primeira velocidade depende da estrutura local da imagem e da geometria do *superpixel* em cada ponto de fronteira. A segunda velocidade calculada depende da distância dos pixels de fronteira da região em expansão a outras regiões, os *superpixels*.

No terceiro passo, Figura 2.3 (c), é realizada a atualização do esqueleto do fundo da imagem, ou seja, em cada iteração é calculado o esqueleto da região da imagem que ainda não foi atingida pela expansão das sementes, para assegurar que as fronteiras dos *superpixels* não vão se cruzar.

O quarto passo, Figura 2.3 (d), é a atualização das velocidades, onde a velocidade de reação-difusão é combinada com um termo adicional de contornos ativos para atrair o fluxo para as arestas. Isso assegura que o limite de evolução diminui quando ele se aproxima de uma região de alto gradiente na imagem.

O quinto passo, ou quarto passo parte b, Figura 2.3 (e), é dado pela atualização de cada pixel na fronteira e de pixels atribuídos ao limite imediato da vizinhança.

O algoritmo termina quando as fronteiras param de evoluir. Após o término do algoritmo é realizado um pós-processamento para deixar as fronteiras dos *superpixels* com apenas um pixel de largura.

2.2 Campos de Markov

Markov Randon Fields (MRF) ou Campos de Markov é uma coleção de variáveis randômicas

$$\Phi = \{\Phi_n : n \in T\},$$

na qual T é um valor enumerável. Nos processamentos em função de T, o futuro do processamento é independente do passado dado, importando somente seu valor presente [31, 20].

Para análise de imagens, Campos de Markov podem ser formulados como um processo de minimização de energia, que é dada por uma função. No entanto, pode acontecer que a solução não seja um mínimo global, pois a função de energia pode ter vários mínimos locais, por ser uma função não-convexa.

Seja

$$X = \{X_i, 1 \leq i \leq N\}$$

o conjunto das variáveis randômicas, onde cada X_i é associado com uma região R_i. E

$$A = \{\lambda_1, ..., \lambda_m\},$$

o conjunto de possíveis rótulos, tais que, $X_i \in A$ para todo i.

Neste cenário, um clique c é definido como um subconjunto de nós de R, de tal maneira, que cada par de diferentes nós em c são vizinhos. Define-se uma função potencial do clique, $p(x)$ envolvendo somente nós c_i de c. Essa função expressa a forma e o grau de interação que cada nó R_i tem com seus vizinhos. De acordo com o teorema de Hammersley-Clifford [3], a distribuição $p(x)$ é dada pela distribuição de Gibbs:

$$p(x) = \frac{1}{Z} \exp\{-\sum_c V_c(x)\} \tag{2.2}$$

onde Z é um fator de normalização, também chamado de *função de partição*. V_c é a função potencial do clique e depende somente das curvas de fluxo que pertencem ao clique c. São utilizados dois pontos do clique, ou seja, assume-se que somente os potenciais não zero são aqueles que correspondem ao um e dois pontos do clique. Cada clique corresponde a um par de fluxos vizinhos. O potencial do clique é dado por:

$$V_c(x) = \begin{cases} -\beta, & x_s = x_q, \quad x_s, x_q \in c \\ +\beta, & x_s \neq x_q, \quad x_s, x_q \in c \end{cases} \tag{2.3}$$

$\beta > 0$ implica que fluxos vizinhos são mais propensos a terem o mesmo rótulo do que

terem rótulos diferentes. Grandes valores de β produzem fluxos suaves. A densidade de probabilidade tem a forma:

$$p(\mathbf{X}|\mathbf{Y}) \propto \exp\left\{p(\mathbf{Y}|\mathbf{X}) - \sum_c V_c(x)\right\} \tag{2.4}$$

Em geral, quando se tem um problema baseado num modelo MRF, a solução procurada geralmente envolve uma estimativa de máximo a posteriori, que é equivalente a um problema de minimização da função de energia E, que é dada por:

$$E = \alpha E_R + \beta \frac{1}{|N(R)|} \sum E_{N(R)} \tag{2.5}$$

na qual α e β são parâmetros definidos empiricamente, E_R é a energia da região R e $|N(R)|$ denota o número de vizinhos de R.

Capítulo 3

Métodos de Monitoramento

Nos últimos anos têm crescido a viabilidade da análise automática de vídeos, devido a melhoria da qualidade dos computadores, câmeras e da qualidade da imagem. Esse fato tem gerado um grande interesse em monitoramento de movimento em vídeo, que é, em geral, um problema desafiador. Mudanças abruptas no movimento, mudanças nos padrões de aparência do objeto e da cena, oclusão objeto sob objeto e objeto sob cena, e o movimento da câmera são grandes dificuldades no monitoramento de objetos. O monitoramento é usualmente realizado num contexto de alto nível de aplicação que requer a localização ou a forma do objeto em todos os quadros. Tipicamente, suposições são feitas para restringir o problema de monitoramento no contexto de uma determinada aplicação.

De acordo com Yilmaz *et al.* [53] existem três etapas principais na análise de vídeos: detecção de objetos de interesse que estão em movimento [40], monitoramento desses objetos quadro a quadro e análise da trajetória do objeto para reconhecer seu comportamento.

De uma forma simples, o monitoramento pode ser definido como um problema de estimar a trajetória de um objeto na imagem e analisar como ele se movimenta no decorrer do vídeo, identificando pontos determinados do objeto nos diferentes quadros do vídeo. Existem várias maneiras, como será abordado mais adiante, de realizar o monitoramento em vídeo. Dependendo do tipo de monitoramento que está sendo realizado, é possível ter informações, tais como, localização do centro do objeto, orientação, área e forma do objeto. Entretanto, o monitoramento de objetos pode ser uma tarefa complexa devido a fatores como perda de informação causada pela projeção de imagens tridimensionais em imagens bidimensionais, ruídos nas imagens, movimentos complexos, natureza articulada dos objetos, oclusão parcial ou total do objeto, mudanças na iluminação da cena e requerimentos para processamento em tempo real.

No contexto de monitoramento em vídeo, objeto é tudo aquilo que esteja em movimento e seja de interesse para análise. Como exemplos, podem ser citados: um navio navegando pelo oceano, carros em uma avenida, pessoas em um parque, bola em um jogo de basquete, entre outros. Os objetos podem ser representados por sua forma e

aparência, utilizando pontos, formas geométricas, formas articuladas, esqueleto, silhuetas e contorno. Tem-se a seguir uma descrição suscinta dessas representações, que também podem ser visualizadas na Figura 3.1, proveniente do trabalho de [53]:

- Pontos - O objeto é representado por seu centróide [49] ou por um conjunto de pontos [45]. A representação de um objeto por um ponto é adequada para monitorar objetos que ocupem pequenas regiões em uma imagem. Na Figura 3.1 pode-se observar dois modelos de representação por pontos, na Figura 3.1 (a) a localização do ponto é o centróide do objeto e em (b) tem-se múltiplos pontos para representar o objeto, que neste exemplo é um corpo humano.
- Formas geométricas primitivas - A forma do objeto é representada por um retângulo, elipse ou outra forma geométrica. As Figuras 3.1 (c) e (d) exibem exemplos de representação do objeto por meio de formas geométricas, onde um corpo humano é representado em (c) por um retângulo e em (d) por uma elipse. Em [34], os autores utilizaram uma representação geométrica adequada para representar objetos rígidos.
- Formas articuladas - Objetos articulados são compostos de partes do corpo que estão dispostas juntamente com as articulações. Como exemplo, tem-se um corpo humano, que é um objeto articulado com torso, pernas, mãos, cabeça e pés conectados por articulações. Para representar esse tipo de objeto podem ser utilizados cilindros e/ou elipses, como mostra a Figura 3.1 (e).
- Esqueleto - Esta representação pode ser utilizada tanto para objetos rígidos quanto para objetos não rígidos. A Figura 3.1 (f) mostra um exemplo de objeto articulado representado por esqueleto. Um exemplo da utilização de esqueletos para representação de objetos pode ser encontrado em [24]. Neste trabalho, partes simétricas de um corpo articulado são detectadas em diferentes escalas por meio do agrupamento de pequenas regiões compactas. Essas regiões podem ser interpretadas como versões deformáveis de discos que compreendem a parte do corpo. A ligação do centro desses discos formam o esqueleto do corpo articulado.
- Silhueta e contorno do objeto - A representação do contorno define o limite de um objeto. A região dentro do contorno é chamada de silhueta do objeto. Representações de silhueta e contorno são adequadas para monitorar formas complexas não rígidas, como é o caso de um corpo humano, que possui várias articulações. Exemplos dessa representação podem ser vistos na Figura 3.1 (g), (h) e (i), onde tem-se respectivamente, os pontos de controle no contorno do objeto, o contorno completo do objeto e a silhueta do objeto.

Diversas abordagens para monitoramento de objetos têm sido propostas na literatura. Elas diferem principalmente no tipo de representação do objeto, nas características da imagem a serem usadas, na aplicação do tipo de movimento e no modelo de aparência.

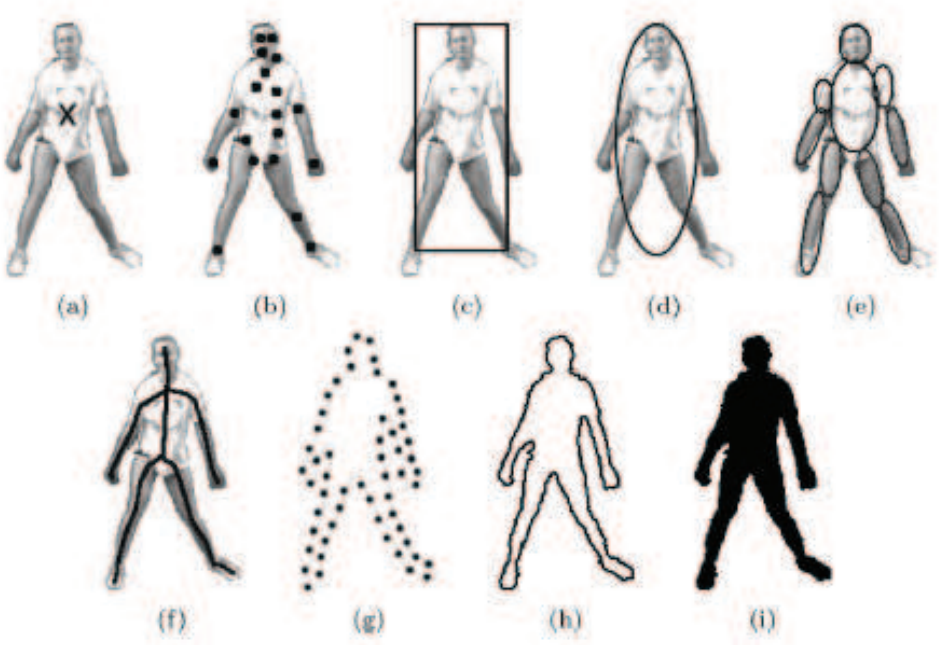


Figura 3.1: Representações de objetos [53]. (a) Centróide, (b) Múltiplos pontos, (c) objeto representado por um retângulo, (d) objeto representado por uma elipse, (e) partes do corpo de um objeto articulado representadas por elipses, (f) esqueleto, (g) pontos de controle no contorno do objeto, (h) contorno completo do objeto, (i) silhueta do objeto.

Todas essas características são designadas de acordo com o contexto e a utilização final do monitoramento que será realizado.

Por essa razão são definidas várias metodologias para monitoramento, que podem ser divididas, entre outras vertentes, em métodos de monitoramento baseados em pontos, baseados em contornos e em modelos de aparência. Essas categorias são sumarizadas a seguir e detalhadas nas seções seguintes:

1. **Métodos de Monitoramento Baseados em Pontos.** Os objetos detectados em quadros consecutivos são representados por pontos e a associação dos pontos é baseada no estado prévio do objeto, que pode incluir posição e movimento do objeto. Esta aproximação requer um mecanismo externo para detectar os objetos em cada quadro [1, 39, 26].
2. **Métodos de Monitoramento Baseados em Silhueta.** Os objetos podem ser localizados por meio de sua silhueta e também podem ser representados por suas delimitações de contorno, sendo estes contornos atualizados dinamicamente em quadros sucessivos. Diversos trabalhos utilizam o contorno para realizar monitoramento, dentre eles [5, 14, 17, 6, 52, 15, 36, 50, 19].
3. **Métodos de Monitoramento Baseados em Modelos de Aparência.** Características de alto nível de regiões da imagem são utilizadas para definir o modelo de aparência. Os algoritmos baseados em modelo de aparência realizam o reconhe-

cimento e monitoramento dos objetos extraindo suas características, agrupando-as em características de alto nível por meio do modelo de aparência e então podendo identificar características semelhantes em diferentes imagens. Os objetos são monitorados fazendo a correspondência entre os modelos dos objetos produzidos por conhecimento a priori, como pode ser observado em [10, 11, 55].

A Tabela 3.1 ilustra as categorias de métodos de monitoramento fazendo referência em cada uma delas aos trabalhos relacionados.

Abordagem	**Trabalhos representativos**
Pontos	Métodos determinísticos [1] Métodos probabilísticos [39, 26]
Silhueta	Correspondência da forma [19, 5, 14, 17] Monitoramento do contorno [6, 52, 15, 36, 50]
Modelos de Aparência	Média, Matriz de covariância [10, 11] Histograma de cores [55]

Tabela 3.1: Categorização dos métodos de monitoramento.

3.1 Métodos de Monitoramento Baseados em Pontos

O monitoramento baseado em pontos pode ser expresso como a correspondência de objetos detectados nos quadros e que são representados por pontos. Entretanto a correspondência de pontos passa por alguns problemas, tais como, presença de oclusões, ausência de detecções, entrada e saída de objetos do quadro. Métodos baseados na correspondência de pontos podem ser divididos em duas categorias: métodos determinísticos e estatísticos. Os métodos determinísticos utilizam heurísticas qualitativas e os métodos probabilísticos analisam as incertezas para estabelecer a correspondência entre os pontos.

Monitoramento de pontos são bastante apropriados para monitorar objetos muito pequenos que podem ser representados por um único ponto. Para representar objetos grandes são necessários vários pontos. Ao monitorar um objeto com múltiplos pontos, o agrupamento automático de pontos que pertencem ao mesmo objeto é um problema importante e nem sempre de fácil solução. Isso acontece porque há a necessidade de distinguir entre vários objetos e entre objetos e o fundo da imagem.

Métodos de monitoramento baseados em pontos podem ser avaliados observando se eles geram a trajetória correta do ponto. Dado que são geradas trajetórias corretas, a

performance pode ser avaliada computando as medidas de precisão e revocação (*precision* × *recall*). Neste contexto, as medidas de precisão e revocação podem ser definidas como:

$$\text{precisão} = \frac{\textit{Número de correspondências corretas}}{\textit{Número de correspondências estabelecidas}} \tag{3.1}$$

$$\text{revocação} = \frac{\textit{Número de correspondências corretas}}{\textit{Número de correspondências atuais}} \tag{3.2}$$

onde correspondências atuais denotam as correspondências disponíveis para uma trajetória correta. Adicionalmente, uma comparação qualitativa para os métodos de monitoramento pode ser baseada em suas habilidades para:

- coordenar entradas de novos objetos e saídas de objetos existentes;
- manipular a falta de informações (oclusões);
- prover uma solução ótima para funções de minimização de custo utilizadas para estabelecer correspondência.

3.1.1 Métodos Determinísticos

Métodos determinísticos definem a associação de cada objeto no quadro prévio para um único objeto no quadro corrente usando um conjunto de restrições de movimento. O custo de minimização da correspondência é formulado como um problema de otimização combinatorial. A solução, que é dada pela correspondência um-para-um entre todas as possíveis associações, pode ser obtida por métodos de "busca gulosos" [21]. O custo de correspondência leva em consideração as seguintes restrições:

- que a localização do objeto não pode sofrer grandes mudanças de um quadro para o outro;
- que a velocidade do objeto é limitada e restringe a correspondência na vizinhança do objeto;
- que a direção e velocidade do objeto não muda drasticamente;
- que a velocidade do objeto em uma pequena vizinhança é similar. Essa restrição é aplicada a objetos representados por múltiplos pontos;
- que a distância entre dois pontos quaisquer no objeto permanecerão as mesmas;
- que combinação da primeira restrição, pouca mudança na localização de um objeto de um quadro para outro, e da segunda restrição, pequena mudança de velocidade, também formam uma restrição.

Essas restrições também podem ser usadas no contexto dos métodos probabilísticos.

O trabalho dado por Ali *et al.* [1] utiliza conceitos da teoria de sistemas caóticos para modelar e analisar ações humanas. As ações são representadas pelas trajetórias de pontos localizados nas articulações do corpo humano, podendo ser localizados seis pontos: as duas mãos, os dois pés, cabeça e barriga. As trajetórias são normalizadas com relação ao ponto localizado no torso, resultando em cinco trajetórias por ação. Primeiramente são extraídos o esqueleto e as pontas do esqueleto utilizando operadores morfológicos na silhueta humana. Um conjunto inicial de trajetórias é gerado unindo as localizações extraídas utilizando restrições espaciais e similaridade do movimento. Trajetórias quebradas ou com associações erradas são corrigidas manualmente. A Figura 3.2 mostra um exemplo de trajetórias extraídas com esse método.

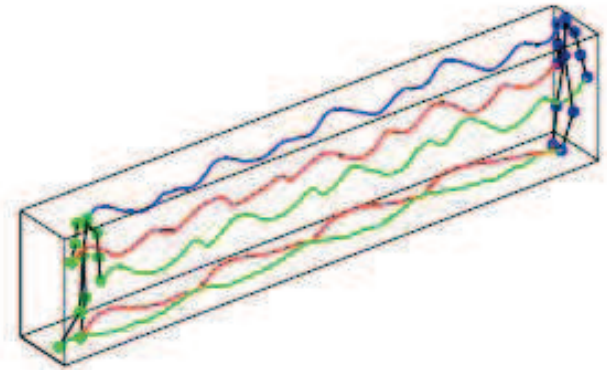


Figura 3.2: Exemplo de trajetórias geradas por pontos localizados nas duas mãos, nos dois pés e na cabeça do humano [1].

3.1.2 Métodos Estatísticos

Métodos estatísticos utilizam a abordagem de espaço de estados para modelar as propriedades dos objetos, que podem ser: posição, velocidade e aceleração. Os pontos usualmente consistem da posição do objeto na imagem, que é obtida por algum mecanismo de detecção. Imagens obtidas de sensores de vídeo inevitavelmente contém ruídos. Métodos estatísticos de monitoramento de pontos explicitamente os manipulam levando em consideração as incertezas do modelo. Essas incertezas podem ser representadas por ruídos normalmente distribuídos. Entretanto, assumir que ruídos são normalmente distribuídos ao redor de suas posições preditas pode não ter uma completa abrangência. Além disso, em muitos casos, os parâmetros do ruído não são conhecidos. Uma outra abordagem para manipular ruídos e a falta de informações é forçar restrições que definem a estrutura tridimensional do objeto. Por exemplo, pode-se forçar os pontos do objeto a se encaixarem na forma tridimensional do objeto.

Em [39], Porikli representa as trajetórias no espaço de características *Hidden Markov Model* (HMM) e determina as trajetórias através de agrupamento dos pontos decompostos

em autovetores, analisando a melhor quantidade de grupos ao invés de usar um número pré-definido deles. Neste modelo é empregado um método de aprendizagem não supervisionado, baseado na decomposição das matrizes de características em autovetores. A trajetória de um objeto é representada como uma coleção de coordenadas da imagem que correspondem ao centro de massa da forma em quadros consecutivos. A forma do objeto pode ser parametrizada por um certo número de variáveis. Elas podem ser as coordenadas superior esquerda e inferior direita no caso da representação de um objeto como um retângulo ou em coordenadas perpendiculares em caso de elipses.

3.2 Métodos de Monitoramento Baseados em Silhueta

O objetivo do monitoramento baseado na silhueta do objeto é encontrar a região do objeto em cada quadro por meio de um modelo gerado utilizando quadros anteriores. Este modelo pode ser um histograma de cores, ou até mesmo o contorno do objeto. Os métodos de monitoramento baseados em silhueta podem ser divididos em duas categorias: correspondência de formas e monitoramento do contorno. Abordagens baseadas na correspondência de formas procuram pela silhueta no quadro corrente. O monitoramento através do contorno, por outro lado, desenvolve um contorno inicial para sua nova posição no quadro corrente utilizando modelo de espaço de estados da cadeia de Markov, onde tem-se o conjunto de valores possíveis para essa nova posição, ou pela minimização de alguma função de energia.

Correspondência de formas e monitoramento do contorno podem ser considerados, essencialmente, como segmentação de objetos aplicado ao domínio temporal utilizando modelos prévios obtidos a partir dos quadros anteriores.

3.2.1 Correspondência de formas

Para encontrar formas correspondentes em diferentes quadros, pode-se computar a similaridade do objeto com o modelo gerado a partir da silhueta do objeto nos quadros anteriores. Nesta abordagem, considera-se que a silhueta do objeto seja apenas translacionada do quadro corrente para o quadro seguinte. O modelo do objeto, que está usualmente na forma de um mapa de arestas, é reinicializado para manipular as mudanças de aparência em todo quadro depois que o objeto é encontrado. Esta atualização é necessária para superar problemas de monitoramento, tais como, mudanças do ponto de visão e das condições de iluminação e movimentos de objetos não rígidos.

No trabalho desenvolvido por Ke *et al.* [19] utiliza-se características volumétricas para reconhecimento de ação, mesmo em ambientes com muitos objetos que não são de interesse. Eles propõem uma representação simples e efetiva, baseada na forma, para alinhar vídeos que não requerem subtração de fundo, combinando essa representação com técni-

cas baseadas em fluxo. O alinhamento baseado na forma consiste da extração da região espaço-temporal e posteriormente do alinhamento dessas regiões. Para a extração da região é utilizada uma técnica de agrupamento não supervisionada para segmentar o vídeo em volumes tridimensionais que mantém a mesma aparência dentro do volume. Esses volumes são chamados de *supervoxels*, mantendo o mesmo conceito de *superpixels* definido por Ren e Malik [42] e discutido no Capítulo 2 dessa dissertação. Normalmente, os limites do objeto nos volumes espaço-temporais correspondem às fronteiras dos *supervoxels*, assim como as arestas dos objetos coincidem com a segmentação dos *superpixels* [32].

Um extrator de região ideal deveria não somente segmentar objetos individuais no espaço, mas também monitorar seu movimento através do tempo. Foi utilizado o algoritmo "*mean shift*" para agrupar o vídeo em regiões, pois segmentar cada imagem do vídeo e depois unir essas regiões temporariamente causaria regiões instáveis. Dessa forma, a segmentação foi realizada empilhando a sequência de quadros do vídeo e depois segmentando o volume tridimensional (espaço e tempo) de pixels. A menor unidade considerada com este conceito é o *voxel*, um pixel 1×1 de um quadro.

Para que os tamanhos das regiões resultantes da segmentação não ficassem iguais, sendo que um tamanho pequeno dividiria grandes regiões em várias partes pequenas e um tamanho grande englobaria várias regiões pequenas, foi definida uma abordagem tal que a imagem é segmentada em uma pirâmide de níveis, na qual cada nível possui um tamanho diferente para as regiões resultantes da segmentação. Na Figura 3.3 tem-se um exemplo com segmentação em quatro níveis, onde o Nível 1 segmenta a imagem com as menores regiões possíveis, o Nível 2 segmenta em regiões um pouco maiores que no Nível 1 e assim por diante até chegar ao nível mais alto, onde tem-se a imagem segmentada em grandes regiões.

Figura 3.3: Diferentes níveis de hierarquia dos tamanhos da segmentação do objeto. Grandes regiões são encontradas em altos níveis de hierarquia [19].

O alinhamento das regiões é feito sobre as diferentes hierarquias. Primeiramente, o algoritmo alinha a forma espaço-temporal do volume, ao invés de alinhar pixels. Depois, o alinhamento é realizado sobre a segmentação espaço-temporal dos volumes, identificando o conjunto de regiões dos *supervoxels* que, quando agregados, melhor se alinham com um dado template.

O algoritmo está baseado na região de intersecção de volumes com apenas duas regiões. Inicialmente, a busca é limitada a um único nível da hierarquia de segmentação e depois

estendido a outros níveis de acordo com a pontuação em cada nível. Se uma região corresponde a várias outras regiões em um determinado nível, então esse nível será penalizado por isso, ficando com maior pontuação a região que melhor se encaixa com a região do template. O template T do volume $|T|$ é deslizado ao longo das dimensões de cada quadro do vídeo e na dimensão do tempo no vídeo. Dessa forma é medida a distância de correlação em todas as localizações no espaço e no tempo. Utilizando limiares e encontrando picos podem ser dadas localizações de alinhamentos potenciais.

Características de fluxo são adicionadas para ajudar a distinguir ações em casos que somente a partir da silhueta não é possível identificar se existe e qual é o movimento. Como é o caso de uma bola girando, sem deslocamento espacial, situação onde não é possível perceber que há movimento. Essa bola é indistinguível em relação a uma bola parada analisando-se somente a sua forma.

Outro método para fazer a correspondência de formas é encontrar silhuetas correspondentes detectadas em dois quadros consecutivos. A detecção de silhueta é geralmente realizada através da subtração do fundo da imagem. Uma vez que as silhuetas são extraídas dos quadros, a correspondência é realizada utilizando alguma distância entre os modelos dos objetos associados com cada silhueta. Os modelos dos objetos usualmente, são da forma de modelos de aparência (cor ou histogramas), contorno da silhueta (contornos abertos ou fechados), arestas do objeto ou uma combinação desses modelos.

Em [5] e [14] movimentos humanos são considerados como formas tridimensionais criadas a partir das silhuetas no volume espaço-temporal do vídeo. Dessa maneira, em uma sequência de imagens de um vídeo, uma ação humana gera uma *forma espaço-temporal* no volume espaço-temporal (Figura 3.4), que é obtida pela concatenação das silhuetas bidimensionais no volume espaço-temporal. Formas espaço-temporais contém informações sobre a pose do humano em qualquer tempo (localização e orientação do torso e membros, relação de aspecto de diferentes partes do corpo) e informações dinâmicas (movimento global do corpo e movimento dos membros em relação ao corpo), considerando superfícies fechadas. Também fazem uso das propriedades do vídeo para extrair características espaço-temporais, tais como, saliências espaço-temporais (parte central de um corpo humano, torso e uma coleção de partes articuladas em movimento), dinâmica da ação, estrutura da forma e orientação local das diferentes partes do corpo.

Em contraste com a busca por possíveis correspondências de silhuetas em quadros consecutivos, o monitoramento de silhueta pode ser realizado computando vetores de fluxo para cada pixel dentro da silhueta, tal que o fluxo que é dominante sobre toda a silhueta é usado para gerar a trajetória da silhueta. Seguindo esta observação, Sato e Aggarwal [44] propuseram o monitoramento de objetos aplicando a transformada de Hough no espaço de velocidade para as silhuetas do objeto em quadros consecutivos. Silhuetas de objetos são detectadas usando subtração do fundo da imagem. Dessa forma, a partir de uma janela ao redor de cada região do pixel em movimento, a transformada

Figura 3.4: Forma espaço-temporal das ações "pular", "caminhar" e "correr" [14].

de Hough é aplicada para calcular as matrizes para o fluxo vertical v e o fluxo horizontal u. Essas matrizes proveêm a Velocidade Espaço-Temporal (TSV) da imagem em 4D (x, y, u, v) para cada quadro, encontrando regiões com padrões similares de movimento, promovendo uma correspondência baseada no movimento da silhueta do objeto.

Jiang e Martin [17] consideram que uma ação pode ser caracterizada pelo movimento e deformação de uma forma. O movimento é representado por uma linha de fluxo (*flow line*), que é uma linha formada pelo monitoramento do objeto através dos quadros que formam o vídeo. A junção de várias linhas de fluxo formam o fluxo da forma (*shape flow*) do objeto, que representa além do movimento do objeto, a sua forma e deformação no decorrer do vídeo. O *shape flow* é utilizado para representar a ação.

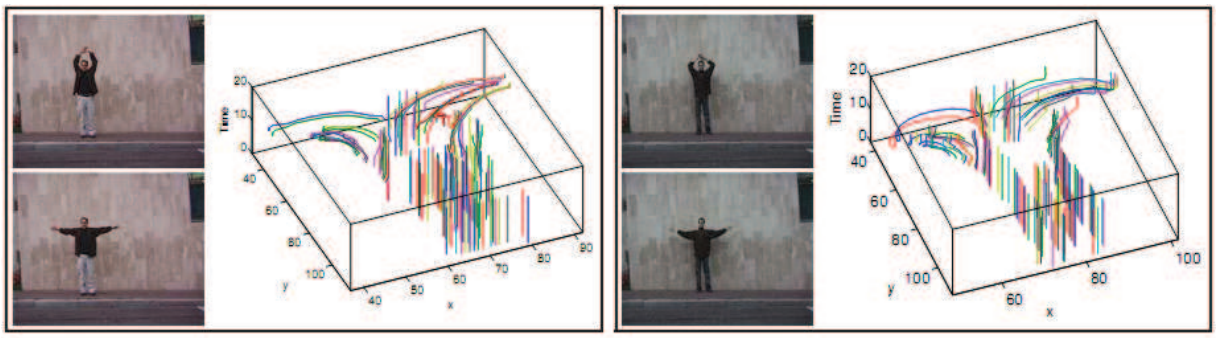


Figura 3.5: Fluxo da forma [17].

A Figura 3.5 exibe dois exemplos de fluxos da forma. Jiang e Martin [17] utilizam um template de fluxo da forma, fazendo a correspondência entre o template e o fluxo da forma sendo analisado, por meio de programação linear [16], para identificar movimentos em vídeos. Procura-se a representação do movimento e uma representação não paramétrica da forma.

Para encontrar os pontos do contorno do objeto é utilizado o detector de Canny e com a triangulação de Delaunay sobre esses pontos são encontradas as relações de vizinhança. O resultado é um campo de movimento esparso, mas ao realizar a interpolação através dos pontos adquiridos por Delaunay, o resultado é um campo denso de movimento. Dessa forma, as linhas de fluxos são formadas fazendo a concatenação dos campos de movimento

quadro a quadro.

As linhas de fluxo são adquiridas baseando-se em *Iterative Conditional Modes* (ICM) [4], estimando pontos de movimento entre quadros adjacentes. O ICM computa vetores de movimento que minimizam o custo de correspondência e maximizam a consistência dos vizinhos. Isso se deve ao fato de que os pixels preservam as seguintes características: as regiões seguem uma distribuição estatística padrão e pixels próximos tendem a ter cores semelhantes.

Após encontrar as linhas de fluxo é necessário alinhar os fluxos da forma para detectar a ação. O domínio de busca é um espaço de tempo no vídeo que tenha a mesma extensão do template da ação, no qual é necessário encontrar um alinhamento consistente de linhas de fluxo entre o template e o vídeo objetivo, sendo que cada ação tem um template que a define. Considera-se que haja pouca mudança na rotação entre o template e o vídeo em análise, mas pode ocorrer uma grande mudança na escala. Dessa forma, o alinhamento de fluxos da forma é considerado como um problema de otimização. O custo do alinhamento é dado pela distância euclidiana entre os dois vetores formados pela normalização das linhas de fluxo em coordenadas bidimensionais.

3.2.2 Monitoramento do contorno

Métodos de monitoramento do contorno, em contraste aos métodos de correspondência da forma, iterativamente evoluem de um contorno inicial no quadro anterior para sua nova posição no quadro seguinte. Essa evolução do contorno requer que alguma parte do objeto no quadro corrente se justaponha com a região do objeto no quadro anterior. Duas maneiras são utilizadas para realizar o monitoramento do contorno: a primeira delas é o monitoramento utilizando modelos do espaço de estados e a segunda é minimizar a energia do contorno utilizando técnicas de minimização direta, tal como gradiente descendente.

No monitoramento utilizando modelos do espaço de estados o estado do objeto é definido em termos da forma e dos parâmetros de movimento do contorno. O estado é atualizado em cada instante de tempo de forma que a probabilidade a posteriori do contorno seja maximizada. A probabilidade a posteriori depende do estado a priori e da semelhança dos contornos, que é usualmente definida em termos da distância do contorno a partir das arestas observadas.

No monitoramento por minimização da energia do contorno, define-se a energia do contorno em termos da informação temporal na forma de gradiente temporal (fluxo óptico) ou estatísticas de aparência geradas a partir do objeto e das regiões do fundo da imagem (quadro). O monitoramento do contorno utilizando gradientes é motivado pelo trabalho de calcular o fluxo óptico, que é derivado da constância de brilho em diferentes quadros. Uma vez que os vetores de fluxo estão em mãos, a energia do contorno, que é baseada na intensidade de brilho em cada quadro, é avaliada. Esse processo é iterativamente realizado

até que a energia seja minimizada.

Uma alternativa ao uso de fluxo óptico é explorar a consistência da estatística computada dentro e fora da região do objeto de um quadro para o outro. Esse método requer a inicialização do contorno no quadro corrente com sua posição prévia.

Brox *et al.* [6] propõem o monitoramento de poses utilizando o contorno e o uso de fluxo óptico para melhorar a correspondência de contornos entre quadros. A extração do contorno é realizada através de segmentação. Primeiramente é obtido um contorno inicial e este contorno evolui para que torne-se ótimo em relação ao modelo de energia utilizado. No modelo de energia definido a probabilidade a posteriori de pertencer à região atribuída é maximizada, ou seja, pontos são atribuídos para a região onde eles se encaixam melhor. Para um ponto se ajustar bem a uma região, seu valor deve se ajustar bem à função de densidade probabilidade dessa região. As probabilidades de objetos e da região do fundo da imagem são modeladas como densidades Gaussianas. Elas podem ser estimadas dado um contorno preliminar e são sucessivamente atualizadas quando o contorno evolui.

Para fazer a correspondência do contorno com a superfície do objeto, primeiramente são determinados os pontos da superfície, que são parte da silhueta do objeto. A projeção de cada um desses pontos é encaixada ao ponto mais próximo do contorno extraído. Depois que a pose é estimada, um novo contorno é calculado e o processo continua iterativamente até a pose convergir, ou seja, até que não existam mais modificações. No entanto, esse método é suscetível a ótimos locais. Para amenizar esse problema são definidas poses iniciais através dos pontos vizinhos e é utilizado o resultado com erro mínimo.

3.3 Métodos de Monitoramento Baseados em Modelos de Aparência

Modelos baseados na aparência tem sido bastante utilizados devido à sua simplicidade e ao baixo custo computacional. Modelos de aparência podem ser representados de diversas maneiras. A seguir, serão apresentadas algumas formas de representar as características de aparência de objetos.

- Densidade de probabilidade - Estima que a aparência do objeto pode ser tanto paramétrica, tal como a Gaussiana ou mistura de Gaussianas, quanto não paramétrica, tal como janelas de Parzen e histogramas. A densidade de probabilidade das características de aparência do objeto, como cor e textura, podem ser computados à partir de regiões da imagem especificadas pelos modelos de forma.
- Templates - São formados usando formas geométricas ou silhuetas. Uma vantagem do template é que ele contém tanto informação espacial quanto informação da aparência.

- Modelo de aparência ativo - São gerados pela modelagem simultânea da forma e aparência do objeto. A forma do objeto é definida como um conjunto de marcações. Essas marcações podem estar nos limites do objeto ou dentro da região do objeto. Para cada marcação, um vetor de aparência que pode conter cor, forma, textura e magnitude do gradiente é armazenado. Modelos de aparência ativos requerem uma fase de treinamento onde a forma e sua aparência são aprendidas a partir de um conjunto de amostras.
- Modelo de aparência de múltipla visão - Esses modelos codificam diferentes visões de um objeto. Uma maneira de representar diferentes visões de um objeto é criar um sub-espaço a partir de cada visão dada. Outra aproximação é aprender diferentes vistas de um objeto, treinando um conjunto de classificadores.

Abordagens baseadas em modelos de aparência incluem um alto nível de conhecimento sobre os dados por meio de um modelo aprendido previamente, normalmente com características extraídas das subregiões da imagem centrada em pontos de interesse espaço-temporais. Por esta razão, a performance desses métodos depende fortemente tanto da escolha do modelo quanto da disponibilidade da informação a priori.

Filipovych e Ribeiro [10] apresentam um método para descrever movimento humano através de um *framework* que codifica os relacionamentos espaço-temporais entre partes descritivas do movimento e a aparência das poses individuais. São utilizadas características, tais como, posição temporal das poses, média e matriz de covariância. O método automaticamente aprende modelos de poses estáticas e partes do movimento espaço-temporal. Este trabalho é desenvolvido para um algoritmo para aprendizagem de estrutura de modelos Bayesianos em espaços de projeção múltipla [11], no qual, modelos de projeções individuais podem ser relacionados através da distribuição de probabilidades.

Zhao *et al.* [55] utilizam um *framework* Bayesiano baseado na aparência de humanos, na visibilidade do corpo e na separação do *background/foreground*. Utilizam o conhecimento de vários aspectos, tais como, contorno, modelo da câmera e características da imagem para formarem o *framework* probabilístico.

Para desenhar as trajetórias são observados o número de objetos, suas correspondências aos objetos nos quadros anteriores, parâmetros (como posição por exemplo) e a incerteza dos parâmetros. É definido um modelo de aparência baseado em cor que considera todo o objeto, juntamente com o fundo e codifica duas premissas: o objeto é diferente do *background* e o objeto é similar ao seu correspondente.

O modelo de aparência do objeto é definido através do histograma de cores, definido dentro da forma do objeto (elipse) que ajuda a estabelecer a correspondência no monitoramento. O histograma de cores é utilizado porque é insensível a não rigidez do movimento humano. Quando o histograma de cores é calculado, uma função Kernel (K_e) é aplicada ao peso da localização do pixel, tal que o centro tenha um peso maior que o da fronteira.

O histograma possui 512 pontos e utiliza o RGB de todas as amostras com as 3 regiões elípticas do modelo do objeto: cabeça, torso e pernas (Figura 3.6).

O modelo de aparência do fundo é uma versão modificada da distribuição Gaussiana. As características média e covariância da cor do pixel são avaliadas e os parâmetros Gaussianos são atualizados constantemente.

Figura 3.6: Elipsóides correspondentes à cabeça, torso e pernas do humano [55].

3.4 Conclusões

Selecionar bem as características é um ponto crucial para o sucesso do monitoramento. Geralmente, é desejável que a característica seja singular para tornar mais fácil a sua identificação no espaço de características a ser pesquisado. A seleção da característica está diretamente ligada à forma de representação do objeto. De fato, se a representação da aparência é feita baseando-se no contorno, então as bordas dos objetos são usadas como características. Se a representação for baseada em histogramas, então uma boa característica será a cor. No entanto, as características podem ser usadas fazendo uma combinação das mesmas.

Muitas características são escolhidas manualmente dependendo do domínio do problema. Entre as características citadas a mais usada para monitoramento é a cor, quando esta não é suficiente ou não é adequada ao problema outras características são incorporadas.

Para o monitoramento baseado em silhueta, algumas limitações para a correspondência espaço-temporal da forma podem ser encontradas. Entretanto, essas técnicas requerem câmeras estáticas e um bom modelo de fundo da imagem. Subtração de fundo gera buracos quando partes do objeto se misturam com o fundo ou cria saliências na silhueta quando sombras fortes estão presentes. Uma limitação mais sutil de técnicas baseadas em silhuetas é que elas ignoram características dentro da fronteira do objeto, tal como o movimento interno do objeto.

Características de fluxo são adicionadas para ajudar a distinguir ações em casos que não é possível identificar que há movimento somente a partir da silhueta. Como é o caso de uma bola girando em seu próprio eixo, onde não é possível perceber que há movimento. O movimento nessa bola é indistinguível em relação a uma bola parada

analisando-se somente a sua forma. Quando uma silhueta é suficientemente detalhada, pode-se identificar o objeto rapidamente ou julgar sua similaridade em relação a outras formas. Técnicas baseadas no fluxo estimam o campo visual entre quadros adjacentes e o utilizam como base para o reconhecimento. Uma vantagem importante em aproximações baseadas em fluxo é que elas não requerem subtração de fundo e podem processar vídeos com limitações do movimento da câmera.

Os movimentos nos vídeos podem ser representados de diversas maneiras, no entanto, o enfoque deste trabalho é representar o movimento utilizando trajetórias. Se as características das trajetórias não forem suficientemente representativas, elas não podem ser usadas para identificar alguns eventos, tais como, início e parada de movimento que requerem uma caracterização mais detalhada. O interesse desse trabalho é desenvolver características mais significativas, gerando um fluxo de trajetórias que representem o movimento. Cada quadro do vídeo é segmentado em *superpixels* e estes são monitorados, via modelo de aparência, para gerar trajetórias que caracterizam o movimento.

Capítulo 4

Identificação de Trajetórias de Movimentos em Vídeo

4.1 Introdução

Em geral, movimentos em vídeos são complexos e articulados. Obter uma representação matemática do movimento, embora seja ideal para o reconhecimento do mesmo, é uma tarefa árdua e desafiadora. Apesar de uma série de desenvolvimentos por parte da comunidade de visão computacional, o reconhecimento de ação ainda é um problema em aberto. Este problema é um importante componente de uma variedade de aplicações, tais como, monitoramento automático de vídeos, interface homem-computador, indexação de vídeo, reconhecimento de gestos e análise de eventos esportivos.

O conhecimento em padrões de movimento tem sido usado para detectar complexidade em movimento de objetos e predizer comportamento, mas também reconhecer eventos de particular interesse em um vídeo. De fato, o movimento de um objeto pode ser descrito de variadas maneiras baseadas em sua forma, silhueta, forma geométrica primitiva, entre outros.

Neste trabalho, é investigada a idéia de representar movimento de objetos utilizando um conjunto denso de trajetórias. Trajetórias descrevem a variação espaço-temporal do movimento em vídeos. A principal idéia por trás desse trabalho está baseada no descritor de fluxo da forma proposto por Jiang e Martin [17]. Eles utilizam a variação no tempo da deformação do contorno da silhueta de uma pessoa para caracterizar uma ação.

Contudo, obter o fluxo da forma baseado no contorno quando o movimento do objeto acontece dentro das fronteiras do contorno do objeto pode ser difícil. Por exemplo, movimentos de humanos contendo a oclusão do objeto pelo próprio objeto irão gerar fluxos da forma complexos. Como é o caso quando um humano está andando e o vídeo mostra imagens da lateral do mesmo, em determinados momentos o braço e mão do lado oposto de visão da câmera ficam ocultos, modificando a forma do objeto (o humano). Dessa

maneira, os fluxos da forma podem não representar bem o movimento. Outro exemplo é o objeto permanecer o mesmo durante um movimento, mas existir rotação na textura do objeto. Fluxos da forma não estarão aptos a descrever a rotação de uma esfera, pois será analisado somente o contorno da mesma, que permanece na mesma posição durante todo o movimento. Neste trabalho, é examinada uma representação de movimento por meio do monitoramento das regiões pertencentes ao objeto em movimento utilizando características de cor dessas regiões, ou seja, serão produzidas trajetórias a partir de informações de dentro e fora do objeto, e não se limitando ao contorno do mesmo.

Neste Capítulo será descrito o método proposto por este trabalho para extrair as trajetórias que representam o movimento em vídeos, bem como as características usadas em cada quadro para segmentar e montar o modelo de aparência de cada região e o processo de extração do modelo de aparência dessas regiões.

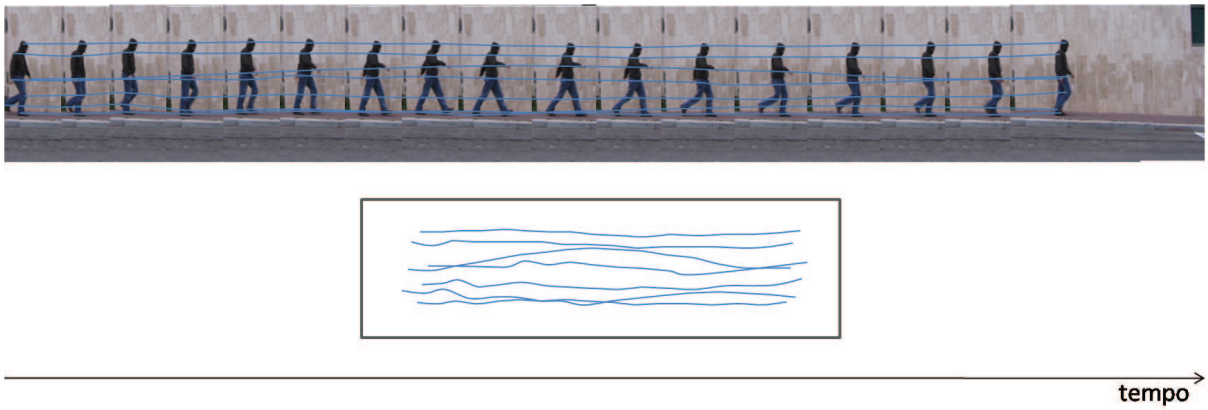
4.2 Modelagem do Problema

A identificação de movimentos por humanos é automática e a identificação do caminho que ele vai seguir é intuitiva [22, 43, 8]. Quando é necessário fazer essa identificação automaticamente por meio de um programa de computador, é preciso utilizar um modelo padrão para a ação para que seja possível identificar modelos semelhantes [27]. Tipicamente, ações em vídeos têm sido definidas por suas características de movimento e descritas por trajetórias [30, 54].

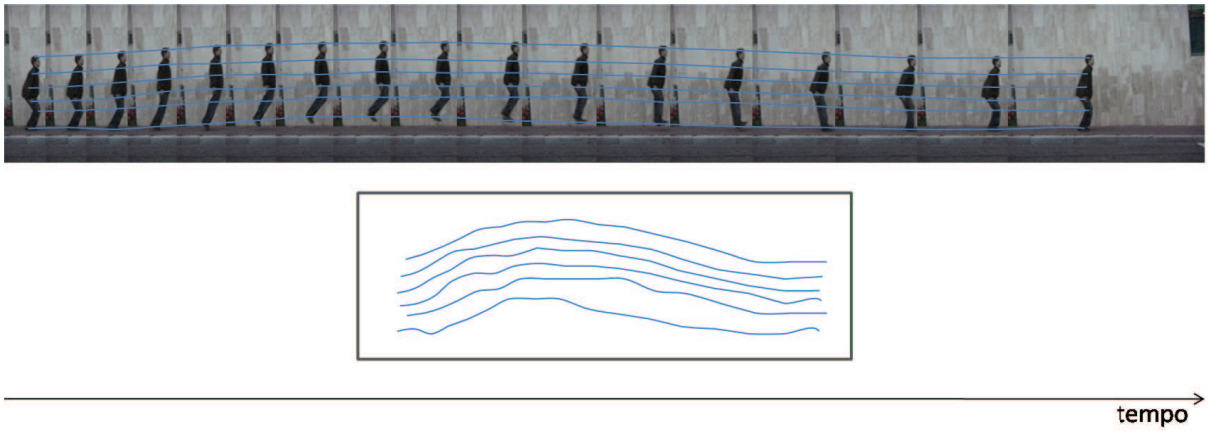
A Figura 4.1 ilustra, através de sequências de imagens de vídeos, como o movimento pode ser representado por trajetórias. Neste exemplo foram escolhidos pontos estratégicos no corpo do ator. O monitoramento desses pontos no decorrer da sequência de imagens no vídeo será responsável por descrever o movimento. A trajetória consiste em identificar automaticamente pontos equivalentes nos quadros seguintes e então fazer a ligação entre os pontos correspondentes, montando as trajetórias.

Vários tipos de movimento, tais como, “andar”, “correr”, “pular”, “andar pulando”, “acenar”, “pular com um pé só”, entre outros, podem ser descritos por trajetórias. A Figura 4.1 (a) mostra um exemplo de trajetórias do movimento “andar”, enquanto a Figura 4.1 (b) exibe um exemplo de trajetórias do movimento “pular”. Observa-se que as trajetórias se diferem de um movimento para o outro. Essa diferença pode ser acentuada com o acréscimo do número de trajetórias utilizadas para representar o movimento.

Para o problema de identificação de movimentos similares em um banco de vídeos, pode-se definir trajetórias que representem o movimento em cada vídeo e então comparar com trajetórias do movimento dado pelo modelo padrão. Um exemplo para ilustrar a semelhança entre as trajetórias que representam um mesmo movimento pode ser observado nas Figuras 4.2 e 4.3. A Figura 4.2 mostra as trajetórias do movimento “andar” para dois atores diferentes. Ao fazer a comparação das trajetórias, é possível perceber como elas



(a)



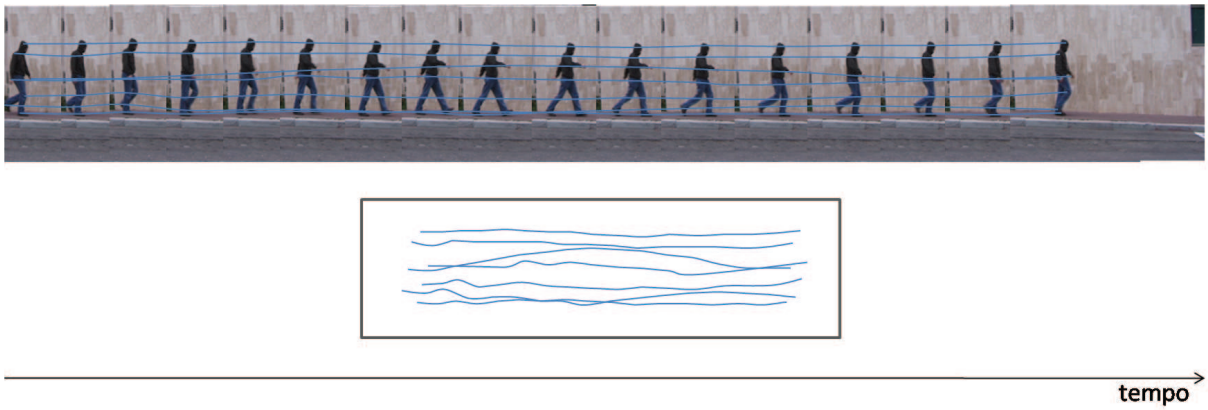
(b)

Figura 4.1: Exemplo de trajetórias que representam movimento em vídeos. (a) Sequência de quadros do movimento "andar" e (b) Sequência de quadros do movimento "pular".

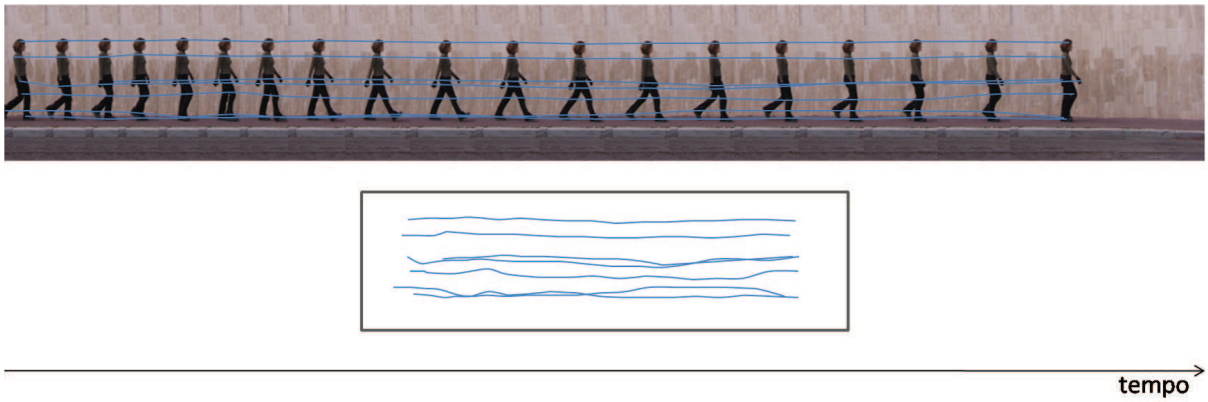
são semelhantes. A Figura 4.3 mostra as trajetórias do movimento "pular com um pé só" para três atores diferentes. Nota-se que mesmo com pessoas diferentes realizando o mesmo movimento, as trajetórias se assemelham.

Como dito anteriormente, ao adicionar trajetórias para representação do movimento, pode-se ter a diferença acentuada de trajetórias para os diferentes tipos de movimento. Ao aumentar o número de trajetórias, obtém-se um fluxo do movimento representado por várias trajetórias. A Figura 4.4 (d) mostra um exemplo de fluxo do movimento de uma parte do corpo de um ator representado por várias trajetórias.

As trajetórias do movimento dos objetos em vídeo podem ser obtidas via análise do movimento [7]. Nesse trabalho, essas trajetórias são representadas pelas coordenadas bidimensionais dos pontos monitorados em cada quadro do vídeo. Trajetórias podem também dar informações de variação temporal quadro a quadro de cada vídeo, sendo então formadas pelo terno: (x,y,t), onde x e y são coordenadas no quadro e t é o quadro do vídeo. A Figura 4.5 ilustra como a terceira coordenada representa a variação temporal indicando a posição t do quadro na sequência de imagens do vídeo, ou seja, uma trajetória



(a)



(b)

Figura 4.2: Sequência de imagens do vídeo com movimento "andar". (a) Primeiro ator e (b) Segundo ator.

pode representar a variação espaço-temporal do movimento.

O problema de modelar o movimento em vídeo utilizando a representação por trajetórias encontra algumas adversidades. Dentre elas estão: a questão de definir os pontos que guiarão as trajetórias; a permanência desses pontos através dos quadros, pois pode ocorrer que haja oclusão de partes do objeto que está em movimento ou até mesmo de todo o objeto, podendo ser uma oclusão objeto sob objeto ou objeto sob cena e o movimento da câmera, pois um mesmo movimento pode ter diferentes trajetórias de acordo com a posição da câmera.

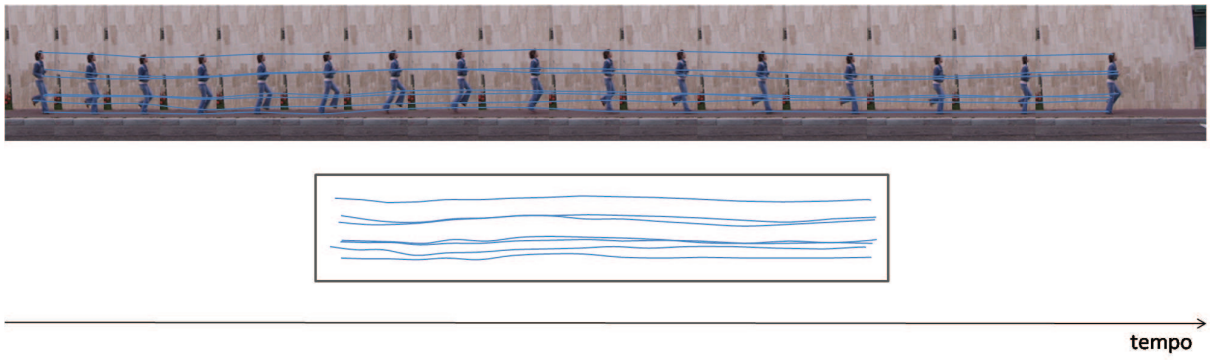
Na representação dos movimentos pode-se usar trajetórias definidas pelo centro de massa; trajetórias de pontos estratégicos do objeto, como por exemplo as articulações em objetos articulados ou por várias trajetórias a partir de pontos espalhados por toda a região do objeto em movimento, como é o caso deste trabalho. Para a identificação de trajetórias, esse trabalho propõe um modelo de representação das regiões por meio dos centróides de cada região em cada quadro do vídeo, e por meio do monitoramento desses pontos, via modelo de aparência, gerar o fluxo do movimento representado pelas

trajetórias das regiões que apresentam movimento. Estes fluxos estarão aptos a descrever a variação espaço-temporal de pontos do objeto. Desta maneira é possível monitorar os objetos em movimento em um vídeo por meio das trajetórias.

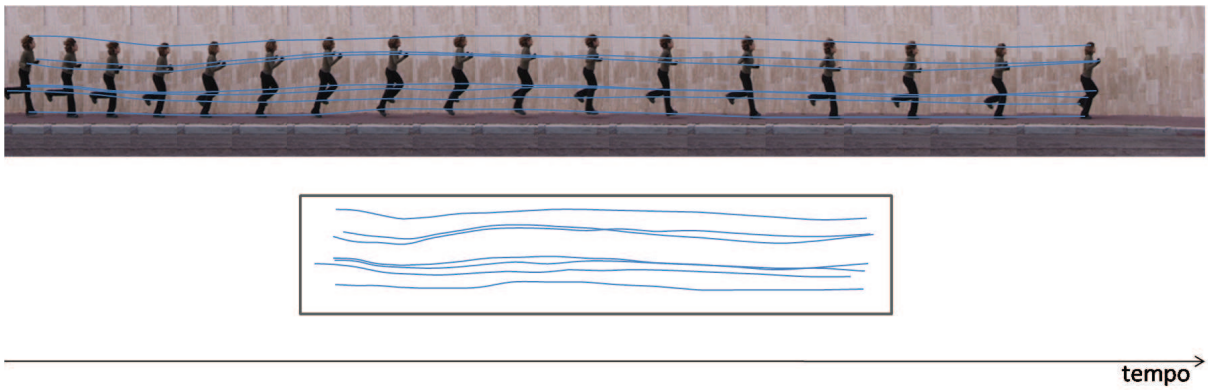
O monitoramento de objetos é importante porque ele torna possível várias outras aplicações, tais como:

- segurança e fiscalização, para reconhecimento de pessoas, promovendo melhor sensação de segurança utilizando informação visual e realizando o monitoramento automático da segurança;
- terapia médica, para melhorar a qualidade de vida de pessoas que fazem terapia física e pessoas com deficiência, por meio do reconhecimento de gestos;
- ajudar na disposição de produtos em lojas, analisando o comportamento dos consumidores para melhorar a organização do ambiente, colocando determinados produtos em posições estratégicas favoráveis à compra;
- índice de vídeos, para obter anotação automática de vídeos, gerando resumos baseados nos objetos;
- recuperação em banco de dados de vídeos através da análise de trajetórias;
- gerenciamento de tráfego, para analisar o fluxo e detectar acidentes;
- edição de vídeos, auxiliando no design futurista de efeitos em vídeos;
- jogos interativos melhorando a interface homem-computador, tornando possível maneiras naturais de interação com sistemas inteligentes;
- análise de eventos esportivos, entre outros.

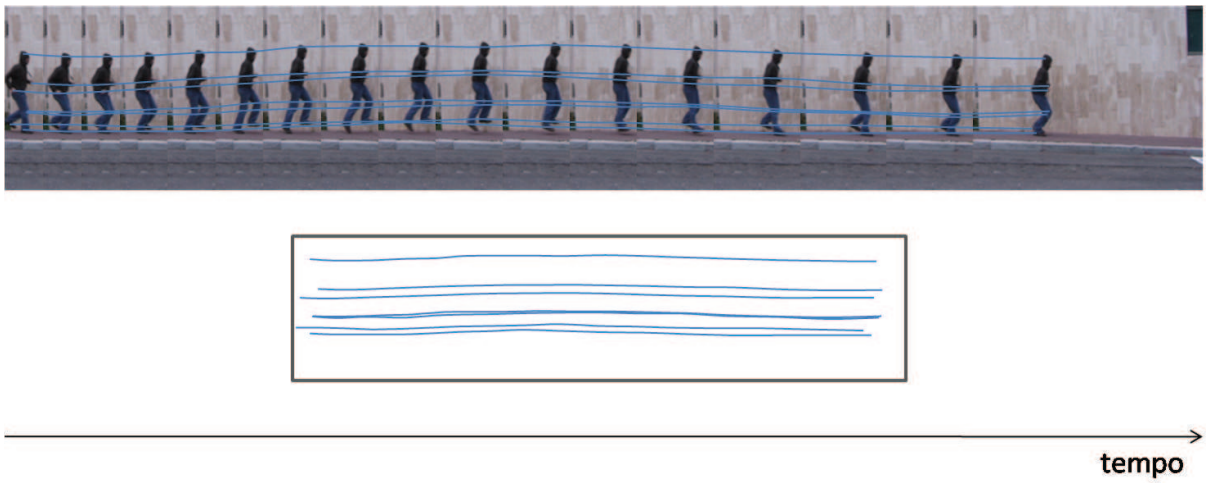
Comumente, o monitoramento requer a localização ou a forma do objeto em todos os quadros. No entanto, a abordagem deste trabalho não requer a identificação da forma do objeto, porque utiliza as regiões que estão em movimento e desta maneira, identifica-se o movimento. Cada quadro do vídeo é segmentado em *superpixels* e estes são monitorados, via modelo de aparência, para gerar trajetórias que caracterizam o movimento. O algoritmo *Iterated Conditional Modes* (ICM), que é baseado em Campos de Markov, é utilizado para analisar a similaridade entre os pontos monitorados, avaliando a aparência da região e a consistência da vizinhança para direcionar o fluxo.



(a)



(b)



(c)

Figura 4.3: Sequência de imagens do vídeo com movimento "pular com um pé só". (a) Primeiro ator; (b) Segundo ator e (c) Terceiro ator.

(a) (b) (c) (d)

Figura 4.4: Exemplo de fluxo de movimento. (a) e (b) Dois quadros do movimento "andar"; (c) Objeto em movimento. (d) A curva vermelha representa a estimativa do fluxo. As trajetórias amarelas representam os fluxos dos vizinhos.

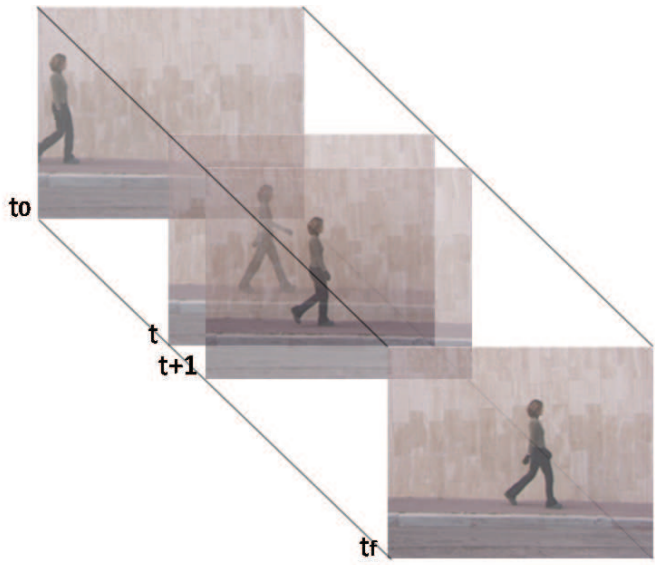


Figura 4.5: Variação temporal da sequência de imagens do vídeo.

4.3 Modelagem Proposta

No estudo de Jiang e Martin [17], fluxos são criados em pontos localizados somente nas arestas do objeto, ou seja, é encontrado o fluxo da forma via monitoramento dos pontos da borda. O monitoramento dos pontos de borda de um objeto pode ser problemático para representar o movimento do mesmo devido à incerteza da localização dos pontos ao longo do contorno e devido a deformação do contorno dependendo do tipo de movimento. A proposta deste trabalho é modificar esta representação incluindo pontos detectados dentro do contorno do objeto em movimento e análise da consistência da vizinhança para representar o movimento do mesmo. O objetivo final é descrever o movimento de um objeto por meio de um conjunto denso de trajetórias obtidas pelo monitoramento de centróides de *superpixels*.

Para atingir os objetivos da abordagem desse trabalho são utilizadas as características da imagem para realizar o monitoramento do movimento por meio da identificação de trajetórias espaço-temporais do movimento. Para tal, têm-se três fases principais de desenvolvimento, que são: a fase de segmentação, a fase de caracterização e a fase de estimativa do fluxo espaço-temporal do movimento. A primeira fase consiste em segmentar cada quadro do vídeo em *superpixels*. Após esta segmentação, o *superpixel* é caracterizado como um modelo de aparência e finalmente, as regiões dos *superpixels* são monitoradas quadro a quadro utilizando o custo de correspondência das regiões correspondentes. Nas Subseções seguintes serão descritos os detalhes dessas fases. A Figura 4.6 apresenta uma visão geral do sistema proposto e na Seção 4.3.4 é dada a síntese do algoritmo.

4.3.1 Fase 1: Segmentação

Após obter a sequência de imagens (quadros) que compõem o vídeo, a primeira fase do trabalho consiste em segmentar cada quadro em *superpixels*, obtendo assim todas as regiões no vídeo passíveis de monitoramento. A segmentação em *superpixels* usada, inicialmente, neste método é a mesma proposta originalmente em [42] e a implementação[1] é uma versão utilizada em [33, 32]. Nesta segmentação, um *superpixel* é um agrupamento local de pixels com características comuns entre os elementos agrupados, além disso, são homogêneos em forma e tamanho.

De acordo com a definição de *superpixels* proposta por Ren e Malik [42], um *superpixel* deve conter elementos que preservem textura e brilho e que no interior do mesmo tenha baixa energia do contorno. Enquanto que em *superpixels* diferentes devem ser encontradas diferentes textura e brilho e a alta energia do contorno deve estar presente nas delimitações do *superpixel*.

Em Mori *et. al* [33, 32] foi utilizado esta abordagem para o problema de reconhecer

[1] A implementação da segmentação em *superpixels* está disponível em `http://www.cs.sfu.ca/~mori/research/superpixels/`

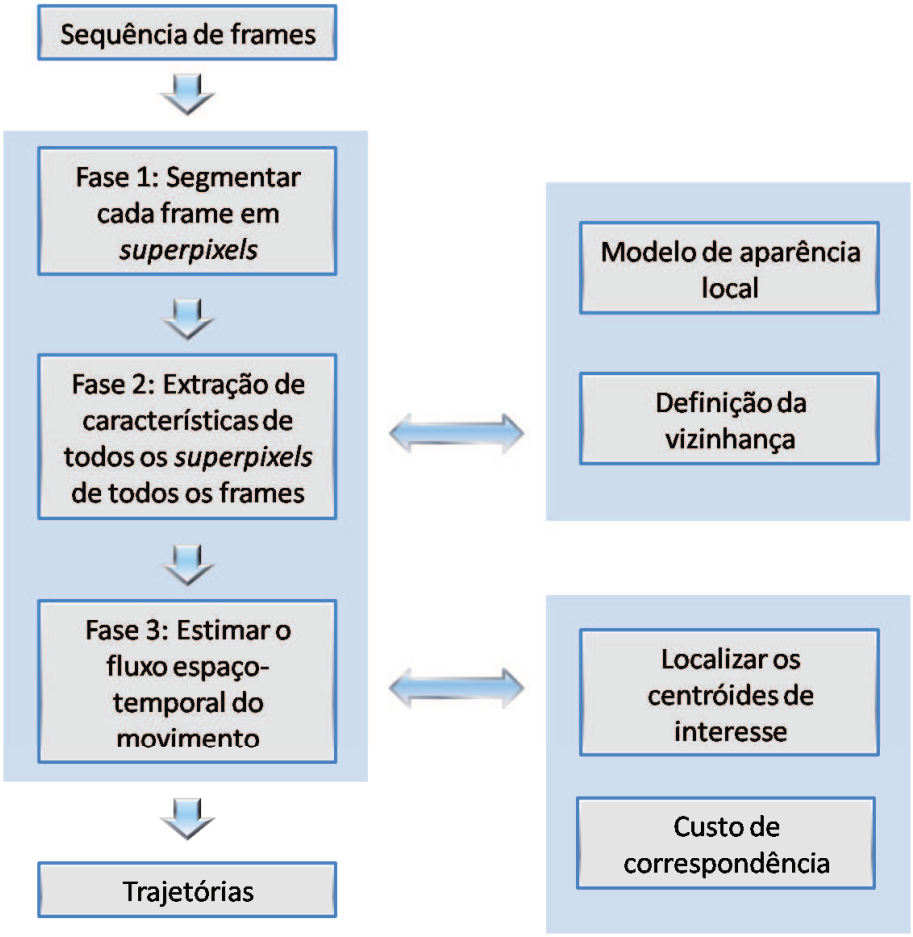
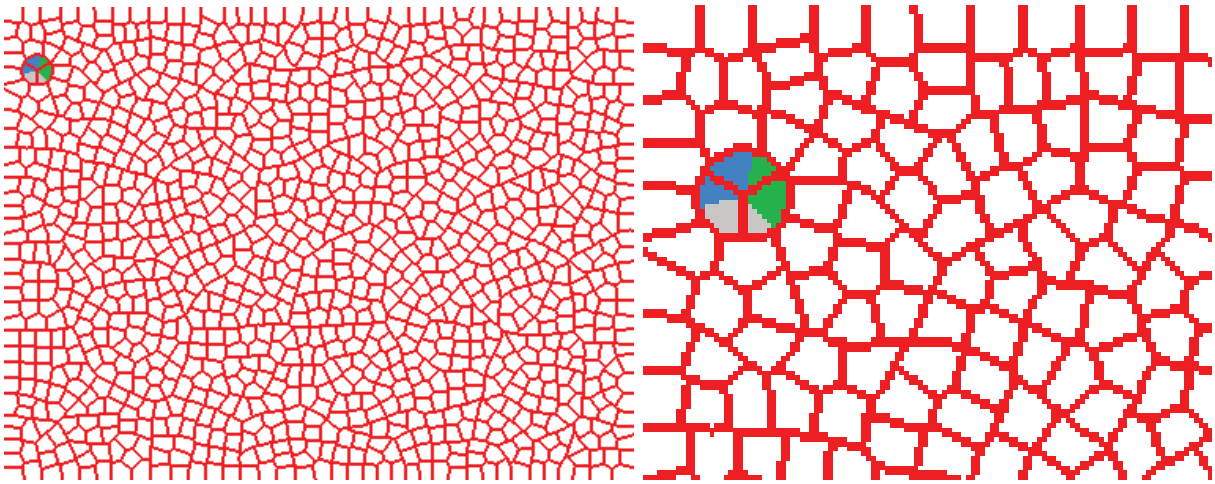


Figura 4.6: Fases do processo de identificação de trajetórias espaço-temporais de movimento em vídeos.

objetos. A segmentação gerou bons resultados de acordo com o objetivo proposto, que era detectar a imagem de um humano em uma figura e localizar suas articulações e membros, em conjunto com as suas máscaras de pixel associadas.

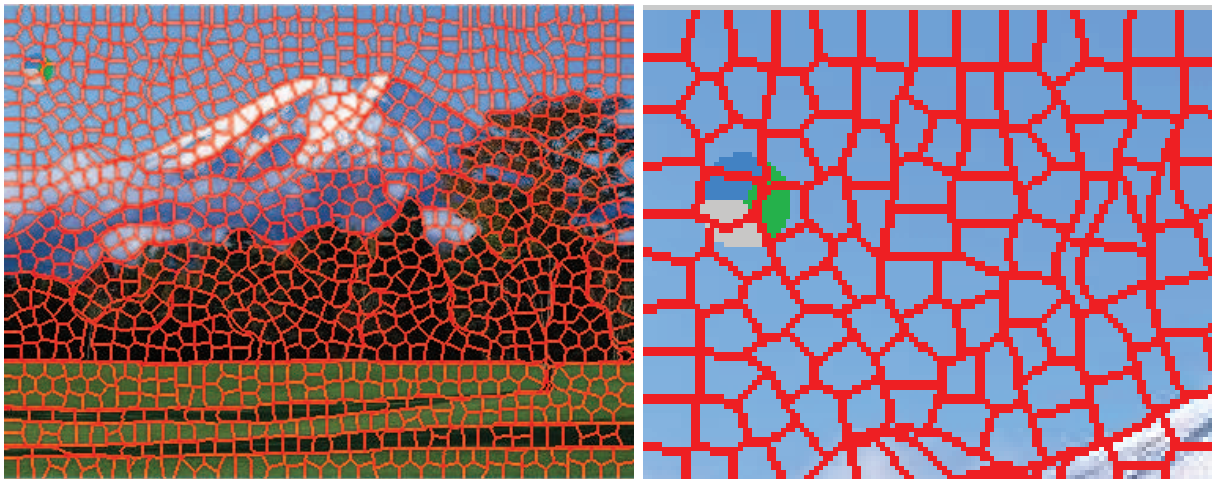
Todavia, para um detalhamento preciso da imagem e quando há necessidade de encontrar objetos desconhecidos na mesma, a segmentação em *superpixels* não trouxe resultados satisfatórios, ou seja, os resultados não foram pertinentes com a proposta de segmentação apresentada. Ao utilizar a segmentação em *superpixels* em imagens sintéticas simples, nota-se a presença de padrões diferentes e alta energia do contorno dentro de um mesmo *superpixel*, como pode ser visto nas Figuras 4.7 e 4.8, onde cores diferentes são agrupadas num mesmo *superpixel* e que regiões com as mesmas características são separadas em *superpixels* distintos, contrariando a definição de *superpixels*.

A Figura 4.7 é a segmentação em *superpixel* de uma imagem contendo uma bola tricolor com fundo branco e a Figura 4.8 é a imagem da mesma bola, onde o fundo é uma paisagem. É importante notar que partes do objeto de uma mesma cor foram segmentadas em diferentes *superpixels*, como é o caso das cores verde, azul e cinza que compõem a bola. Outra questão importante é que diferentes cores foram colocadas em um



(a) (b)

Figura 4.7: (a) Segmentação em *superpixels* de uma imagem contendo uma bola tricolor em um fundo branco; (b) Zoom de (a).



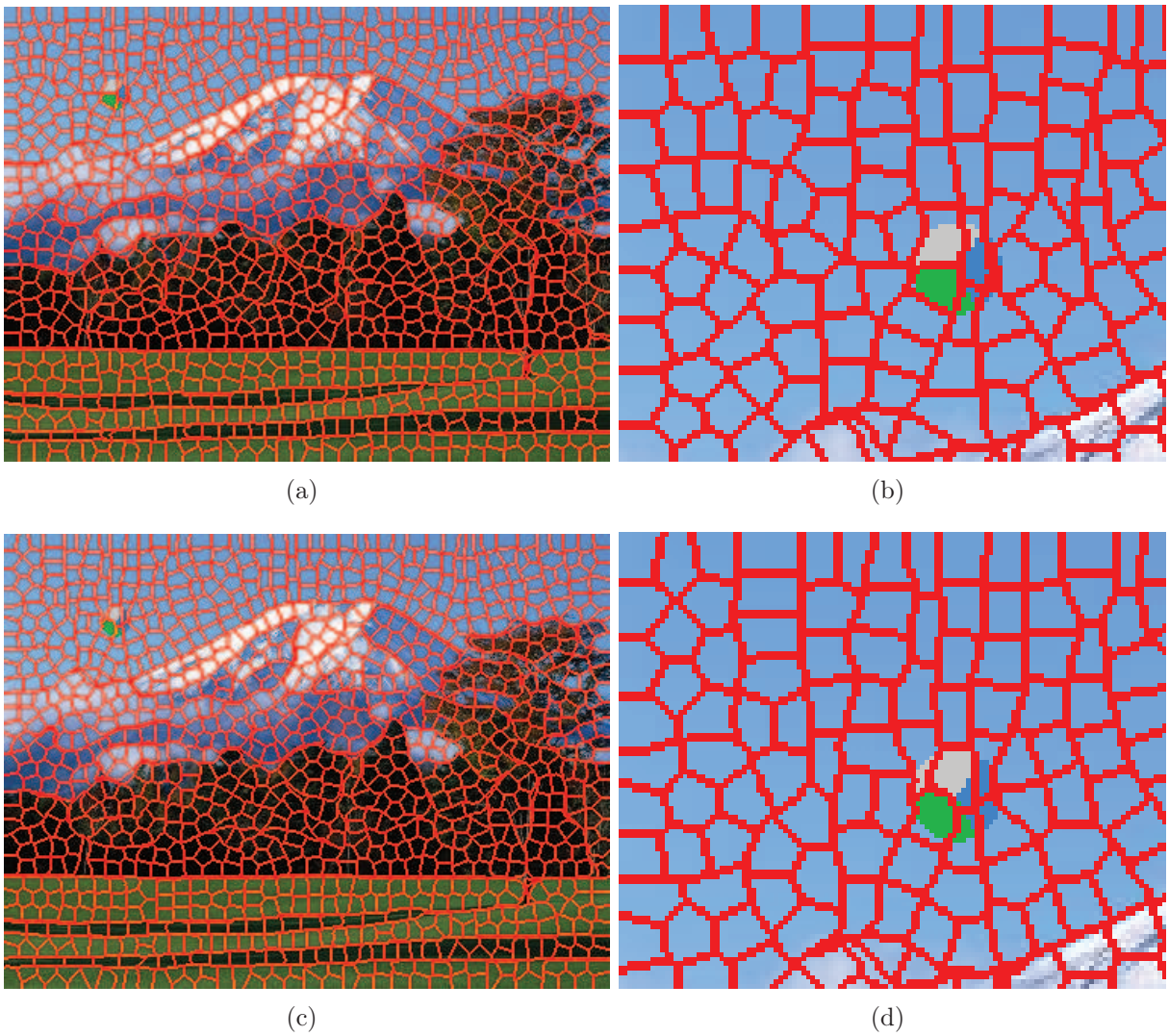
(a) (b)

Figura 4.8: (a) Segmentação em *superpixels* da imagem de uma bola tricolor onde o fundo é uma paisagem; (b) Zoom de (a).

mesmo *superpixel*, como ocorreu na segmentação da bola, onde as cores foram totalmente misturadas em um mesmo *superpixel*, ficando os pares azul e verde, verde e cinza e cinza e azul em um mesmo *superpixel*, contrariando totalmente a definição de *superpixels*, pois há alta energia do contorno dentro da região de um *superpixel*, não existindo a preservação da similaridade intra-região.

O fundo da imagem influencia a segmentação, quando foi utilizado um fundo branco, a bola foi segmentada de forma correta. Com o fundo da imagem mais complexo, nem mesmo os limites da bola foram preservados e continuaram sendo desprezados os conceitos de similaridade intra-região e (dis)similaridade inter-região.

Outra dificuldade encontrada com essa segmentação é que ao executar o programa duas vezes sobre a mesma imagem, diferentes segmentações podem ser obtidas, uma vez que é utilizado um processo aleatório para inicialização da segmentação. Com isso é possível perceber que para quadros sequenciais com poucas diferenças de movimentos, pode-se ter dois quadros com segmentações bastante diferentes, não sendo possível, desta forma, identificar a mesma região em quadros diferentes, embora os quadros sejam muito parecidos. Este fato pode ocasionar a geração de trajetórias distintas para uma mesma coleção sequencial de imagens, já que as trajetórias são definidas de acordo com o monitoramento dos centróides dos *superpixels*, como será melhor explicado posteriormente. A Figura 4.9 mostra o exemplo de dois diferentes resultados da segmentação em uma mesma imagem.



(a) (b)

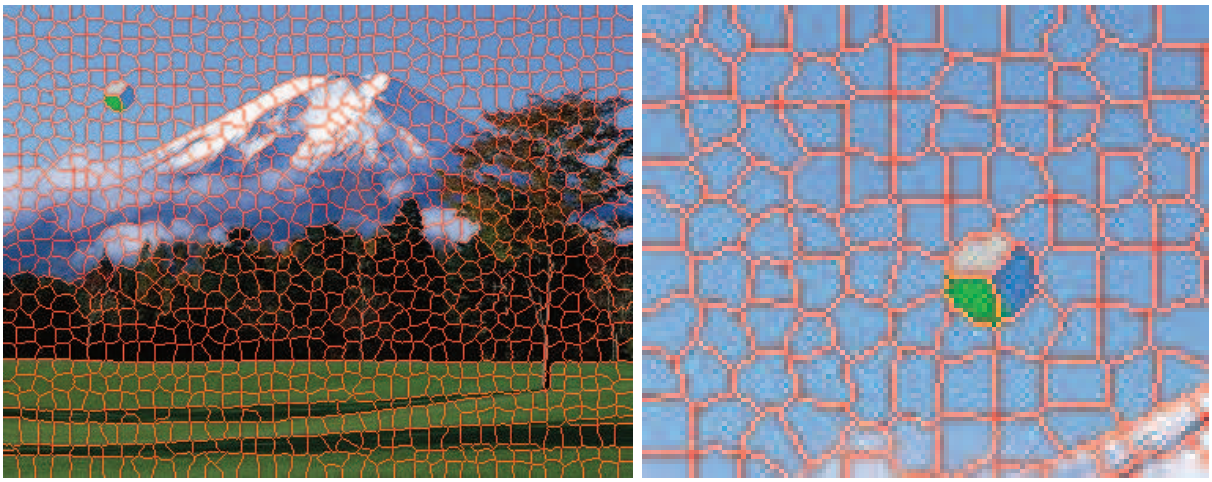
(c) (d)

Figura 4.9: (a) Imagem segmentada em *superpixels*, primeira execução do programa; (b) Zoom de (a); (c) Imagem segmentada em *superpixels*, segunda execução do programa utilizando a mesma imagem; (d) Zoom de (c).

Os problemas encontrados na segmentação não interferiram nos resultados dos tra-

balhos de Mori *et. al* [33, 32], pois não houve a necessidade de segmentar detalhes da imagem. Em seu trabalho um corpo humano é constituído pelos membros braços, pernas e torso. As junções entre os membros são as articulações e são identificadas como uma aglomeração de *superpixels* [32]. Após identificados os *superpixels* que formam os membros, faz-se a representação de cada membro como um retângulo.

Em função dessas anormalidades da segmentação em *superpixels*, foi utilizada nesse trabalho o algoritmo de segmentação em *superpixels* proposto por Levinshtein *et al.* [25], denominado *TurboPixels*. Nesse algoritmo a divisão em *superpixel* é baseada em fluxos geométricos e respeita as fronteiras locais da imagem, como mostra a Figura 4.10.



(a) (b)

Figura 4.10: (a) Imagem segmentada em *superpixels*, utilizando o algoritmo *Turbopixels* [25]; (b) Zoom de (a).

4.3.2 Fase 2: Extração de Características

A segunda fase consiste em utilizar um modelo de aparência local, para definir a característica de cada *superpixel*. Nesta fase são definidos os vizinhos de cada *superpixel*. A relação de vizinhança é dada pela triangulação de Delaunay, dada abaixo.

Cada *superpixel* é representado por seu centróide e as trajetórias do movimento serão formadas pelas localizações desses centróides. Dessa forma, o conjunto de pontos P de cada quadro T é representado da seguinte forma:

$$P^T = \{X_1^T, ..., X_N^T\} \mid X_i^T = (x_i^T, y_i^T) \tag{4.1}$$

onde X_i^T é o centróide do i-ésimo *superpixel* no T-ésimo quadro.

Para cada *superpixel* i no quadro T é necessário ter o modelo de aparência local de X. O modelo de aparência A é definido como segue:

$$A(X_i^T) = \mu_{X\,i} \tag{4.2}$$

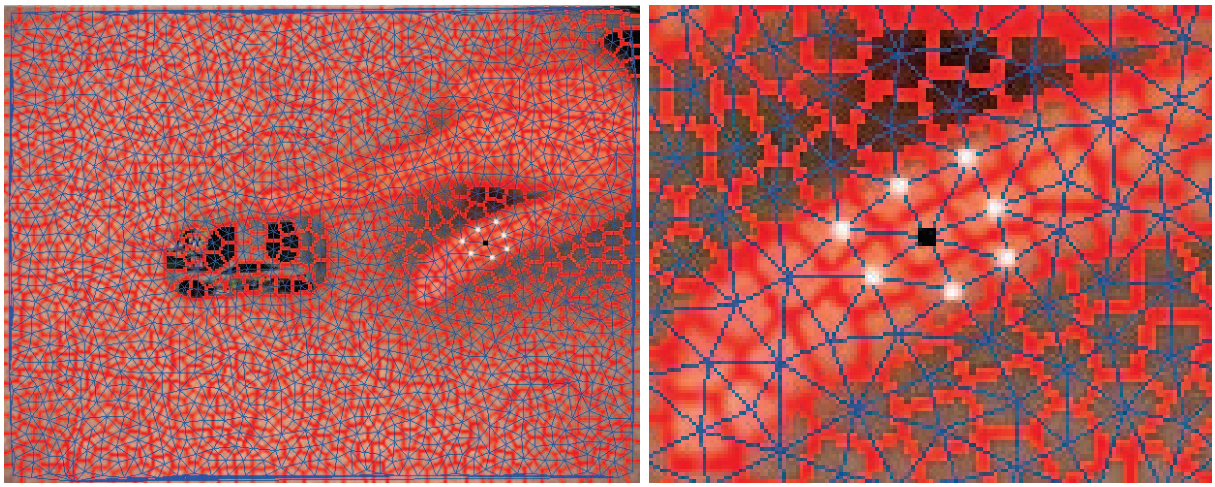
onde $\mu_{X\,i}$ é a média em cada canal de cor RGB de todos os pixels que formam o *superpixel* X_i.

O próximo passo dessa fase é definir a relação de vizinhança dos centróides utilizando a triangulação de Delaunay.

Relação de Vizinhança

Como cada centróide representa o *superpixel* a que pertence, para definir a relação de vizinhança entre os centróides foi utilizada Triangulação de Delaunay cujos vértices são os centróides. Dessa forma são determinados quais são os vizinhos de cada centróide. Os vizinhos de um centróide p são aqueles que estão conectados a ele por uma aresta.

O número de vizinhos é escolhido empiricamente, isto será melhor discutido no capítulo de resultados, Capítulo 5, Seção 5.1.2. O conjunto de vizinhos do centróide $p = p(x)$, $x \in \mathbb{R}^S$, $S = 2$ ou $S = 3$ é representado por $N(p)$. A Figura 4.11 mostra um exemplo de vizinhança com sete vizinhos. Os caminhos em vermelho mostram as delimitações dos *superpixels* e em azul é dada a malha de Delaunay que conecta os centróides. O centróide p que está representado pelo ponto preto na mão tem como vizinhos os centróides representados pelos pontos brancos.



(a) (b)

Figura 4.11: (a) Relação de vizinhança do ponto p, representado pelo ponto em preto na mão e seus vizinhos representados pelos pontos brancos. (b) Zoom de (a).

Correspondência de Vizinhança

Após serem definidas as relações de vizinhança dos centróides é necessário saber como eles podem se corresponder entre quadros diferentes. Foi definida uma representação de permutações para escolher a melhor correspondência entre vizinhos de quadros subsequentes. São analisadas todas as combinações de vizinhos de quadros distintos ao ponto em questão e a melhor correspondência é aquela que apresentar a menor diferença entre os respectivos valores dados pelo modelo de aparência. Como os centróides representam a região do *superpixel*, essa diferença é analisada para todos os pixels da região do *superpixel*.

Para ilustrar vamos considerar que cada centróide tem três vizinhos. Ou seja, considere os pontos $\{p_1, p_2, p_3\}$ vizinhos de p_T no quadro T e sejam $\{q_a, q_b, q_c\}$ vizinhos de q_{T+1} no quadro $T+1$. As combinações são as seguintes:

$$W_1 = \{(p_1, q_a), (p_2, q_c), (p_3, q_b)\} \mid w_1 = |\mu_{p_1} - \mu_{q_a}| + |\mu_{p_2} - \mu_{q_c}| + |\mu_{p_3} - \mu_{q_b}|$$

$$W_2 = \{(p_1, q_a), (p_2, q_b), (p_3, q_c)\} \mid w_2 = |\mu_{p_1} - \mu_{q_a}| + |\mu_{p_2} - \mu_{q_b}| + |\mu_{p_3} - \mu_{q_c}|$$

$$W_3 = \{(p_1, q_b), (p_2, q_a), (p_3, q_c)\} \mid w_3 = |\mu_{p_1} - \mu_{q_b}| + |\mu_{p_2} - \mu_{q_a}| + |\mu_{p_3} - \mu_{q_c}|$$

$$W_4 = \{(p_1, q_b), (p_2, q_c), (p_3, q_a)\} \mid w_4 = |\mu_{p_1} - \mu_{q_b}| + |\mu_{p_2} - \mu_{q_c}| + |\mu_{p_3} - \mu_{q_a}|$$

$$W_5 = \{(p_1, q_c), (p_2, q_a), (p_3, q_b)\} \mid w_5 = |\mu_{p_1} - \mu_{q_c}| + |\mu_{p_2} - \mu_{q_a}| + |\mu_{p_3} - \mu_{q_b}|$$

$$W_6 = \{(p_1, q_c), (p_2, q_b), (p_3, q_a)\} \mid w_6 = |\mu_{p_1} - \mu_{q_c}| + |\mu_{p_2} - \mu_{q_b}| + |\mu_{p_3} - \mu_{q_a}|$$

A melhor combinação é aquela que possui o menor valor para w. Neste exemplo, como são apenas três vizinhos de cada *superpixel*, o número máximo de combinações, dois a dois, é seis.

$$l = \arg \min_{1 \leq i \leq 6} w_i \tag{4.3}$$

O procedimento é análogo para n vizinhos.

4.3.3 Fase 3: Estimativa do Fluxo Espaço-Temporal do Movimento

Na terceira fase calcula-se a estimativa do fluxo espaço-temporal do movimento, onde é necessário localizar as regiões de interesse, ou seja, aquelas que estão em movimento no vídeo. Também é nesta fase que calcula-se o custo de correspondência entre as regiões em movimento nos quadros subsequentes para construção das trajetórias. Para representar o movimento, tem-se como objetivo, construir um conjunto denso de trajetórias conectando *superpixels* similares através dos quadros que descrevem o movimento. Para este propósito são feitas duas suposições. A primeira é que pixels que pertencem a um objeto em movimento não sofrerão mudanças abruptas em sua cor ou brilho de um frame a outro no vídeo. A segunda suposição é que as características da vizinhança de um pixel devem permanecer aproximadamente as mesmas [4] durante todo o vídeo. A Figura 4.12 ilustra a idéia dessas suposições. A aparência do *superpixel* e a estrutura da vizinhança são preservadas durante o movimento do *superpixel* selecionado.

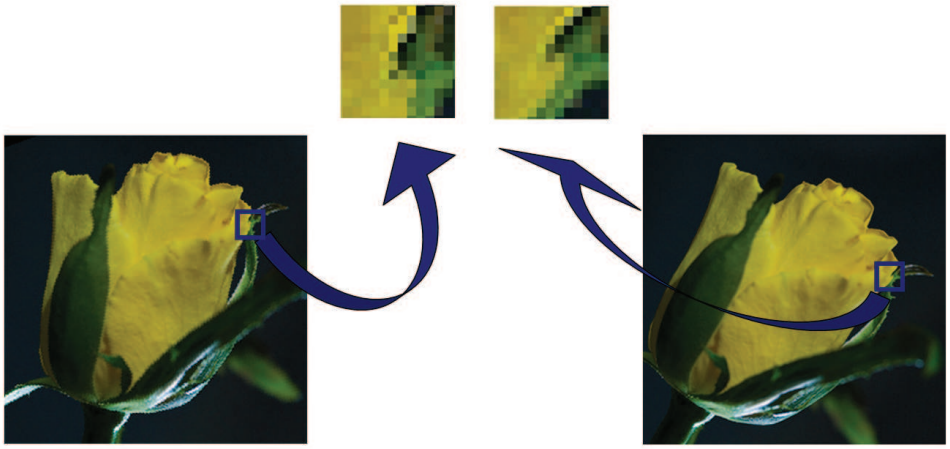


Figura 4.12: Mesma vizinhança em dois quadros diferentes de um vídeo. O objeto em movimento é o mesmo, mas aparece em diferentes posições na imagem. No entanto, tanto a estrutura da vizinhança como a aparência do pixel são preservados dentro da região selecionada.

Baseando-se nessas duas suposições, o algoritmo *Iterated Conditional Modes* (ICM) [4] é utilizado para resolver o problema de encontrar o superpixel no frame posterior que tenha

aparência mais semelhante ao superpixel no frame que está sendo analisado, ou seja, é estimada a probabilidade máxima a posteriori. O ICM é baseado em campos de Markov e está apto a, simultaneamente, estimar os parâmetros de aparência local do *superpixel* e a consistência da vizinhança. Dado um modelo de uma região de pixels em um quadro, o objetivo é encontrar um padrão de pixels similar a este modelo no quadro subsequente. O ponto desafiador desse método é localizar um *superpixel* com característica semelhante, obtida por meio do modelo de aparência, no quadro subsequente. Com o monitoramento do *superpixel* quadro a quadro, o resultado dessa fase é a obtenção das trajetórias que representam o movimento. A Figura 4.13 mostra um exemplo de correspondência ótima entre regiões de duas imagens. Um bom resultado pode ser encontrado ao realizar uma busca exaustiva em uma pequena área do quadro.

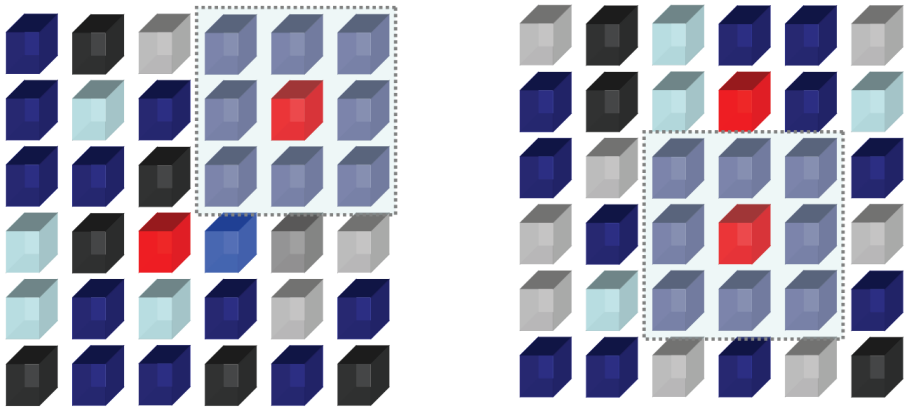


Figura 4.13: Correspondência ótima.

De posse das características de cada *superpixel*, obtidas na segunda fase, é feito o monitoramento de seu deslocamento através dos quadros e é construído o fluxo do movimento usando cada centróide do quadro T, que está na posição X, comparando-o com os n centróides próximos à posição ocupada pelo ponto X no quadro $(T+1)$. Isto é realizado tomando como hipótese que o vídeo possui amostragem suficientemente alta para que o movimento no vídeo seja contínuo, ou seja, um certo centróide não se desloca muito em quadros subsequentes.

O passo seguinte é determinar qual centróide dará continuidade à trajetória que está sendo desenhada. No entanto, vários centróides do quadro $(T+1)$ podem corresponder à região definida pelo centróide no quadro T. A decisão para determinar a continuidade da trajetória, mantendo sua consistência, é ponderada pela consistência da vizinhança, que é dada pelo segundo termo das Equações 4.4, 4.7 e 4.8. Pela equação define-se o custo de correspondência entre os *superpixels* em quadros subsequentes.

O custo de correspondência do ponto X_i no quadro T ao ponto Y_j no quadro $(T+1)$ pode ser obtido por diferentes maneiras. A seguir são apresentadas três opções para o cálculo do custo de correspondência. A primeira é dada utilizando a estimativa da

densidade de probabilidade:

$$C(X_i^T, Y_j^{T+1}) = \alpha\,(1 - \gamma(X_i^T, Y_j^{T+1})) + \beta\ \Sigma_{(p,q)\epsilon W_l}\,(1 - \gamma(p^T, q^{T+1})) \tag{4.4}$$

onde l é dado pela Equação 4.3 e os parâmetros α e β são, respectivamente, o peso de correspondência *superpixel* com *superpixel* e o peso de correspondência da vizinhança. Os valores desses parâmetros são muito importantes, pois é através deles que a escolha do próximo centróide que guiará a trajetória será tomada. Sendo que o custo de correspondência é dado pela correspondência centróide a centróide ponderada por α e pela correspondência dos vizinhos ponderada por β. $\gamma(a, b)$ é a estimativa da densidade de probabilidade, determinada por uma aproximação Gaussiana, definida como segue:

$$\gamma(a, b) = \frac{1}{(2\pi)^{d/2}|\Sigma_a|^{1/2}}\, exp\left\{-\frac{1}{2}(\mu_b - \mu_a)^T\, \Sigma_a^{-1}(\mu_b - \mu_a)\right\} \tag{4.5}$$

onde d é a dimensão do espaço de características. μ_b é o vetor média de b, μ_a é o vetor média de a, Σ_a é a matriz de covariância $d \times d$ de a e $|\Sigma_a|$ é o determinante de Σ.

A Equação 4.6 define o modelo de aparência A dos *superpixels* da seguinte maneira:

$$A(X) = \mathcal{N}(\mu_{\mathcal{X}}, \sum_{\mathcal{X}}) \tag{4.6}$$

onde μ_X é o vetor média que contém as médias de cores dos canais RGB, $\sum_X$ é a matriz de covariância.

Outras formas para calcular o custo de correspondência são dadas pelas Equações 4.7 e 4.8.

$$C(X_i^T, Y_j^{T+1}) = \alpha\,(\delta(X_i^T, Y_j^{T+1})) + \beta\ \frac{1}{|N(Y_j)|}\ \Sigma_{(p,q)\epsilon W_l}\,(\delta(p^T, q^{T+1})) \tag{4.7}$$

$$C(X_i^T, Y_j^{T+1}) = \alpha\,(\delta(X_i^T, Y_j^{T+1})) + \beta\ \ (\delta(\frac{1}{|N(X_i)|}\Sigma_{(X_{p_i}\epsilon N(X_i))}X_{p_i}, \frac{1}{|N(Y_j)|}\Sigma_{(Y_{p_j}\epsilon N(Y_j))}Y_{p_j})) \tag{4.8}$$

no qual os parâmetros α e β tem o mesmo significado que nas fórmulas de custo anteriores.

$|N(Y_j)|$ e $|N(X_i)|$ denotam a quantidade de vizinhos de Y_j e X_i, respectivamente. $\delta(a, b)$ é a diferença entre os valores dados pelo modelo de aparência dos *superpixels* representados por a no quadro T e b no quadro $(T + 1)$:

$$\delta(a, b) = |A(a^T) - A(b^{T+1})| \tag{4.9}$$

Para as duas últimas maneiras de calcular o custo, o modelo de aparência é o mesmo dado pela Equação 4.2.

Para encontrar o *superpixel* com menor custo de correspondência foi utilizado a formulação dos Campos de Markov, onde cada quadro do vídeo está segmentado em regiões disjuntas, os *superpixels*, de tal forma que

$$R = \{R_i : 1 \leq i \leq N\}, \tag{4.10}$$

onde R é uma região que cobre todo o quadro, N é o número de regiões (*superpixels*) do quadro e R_i é o i-ésimo *superpixel* do quadro. A partir dessas regiões cria-se um grafo de regiões adjacentes (*Region Adjacency Graph (RAG)* [48]) denotando cada R_i como um nó do grafo e conectando diretamente os nós do grafo se eles possuem uma relação de vizinhança, como visto na seção anterior.

A solução do problema pode ser dada como um problema de minimização de energia, fazendo uma estimativa de mínimo a posteriori. A função de energia utilizada na abordagem desse trabalho utiliza o peso de correspondência entre os centróides analisados, ou seja, a correspondência centróide a centróide e o peso de correspondência da vizinhança. A função de energia E usada é dada por:

$$E = \alpha E_{R_i} + \beta \frac{1}{|N(R_i)|} \sum E_{N(R_i)} \tag{4.11}$$

na qual α e $\beta > 0$ são parâmetros definidos empiricamente, que correspondem respectivamente, ao peso dado a correspondência centróide a centróide e o peso dado a correspondência da vizinhança, E_R é a energia da região R_i e $N(R_i)$ denota o conjunto de vizinhos de R_i.

Dessa forma, o prolongamento da trajetória que passa por X_i^T é X_k^{T+1}, onde

$$k = \arg \min_j \ C(X_i^T, Y_j^{T+1}) \tag{4.12}$$

onde C é uma das equações que definem o custo de correspondência: Equações 4.4, 4.7 e

4.8.

O fluxo de trajetórias obtido é composto pelas trajetórias dos pontos monitorados. Os movimentos não necessitam estar definidos do começo ao fim do vídeo, mas precisam apresentar um bom grau de homogeneidade espaço-temporal, isto é, o movimento deve seguir o mesmo percurso.

4.3.4 Algoritmo de Identificação de Trajetórias de Movimento

A seguir será descrito o algoritmo utilizado para identificação das trajetórias de movimento em um dado vídeo:

Algoritmo de estimativa do fluxo espaço-temporal

Entrada: Sequência de imagens que compõem o vídeo
Saída: Fluxo de trajetórias do movimento

1: Ler uma sequência de n quadros
2: **for** 1 até n **do**
3: Segmentar cada quadro em *superpixels*, definindo cada região R_i no quadro T
4: **end for**
5: **for** 1 até n **do**
6: **for** 1 até o número de *superpixels* em cada quadro **do**
7: Definir o centróide do *superpixel*. Após esse laço tem-se o conjunto de pontos P do quadro T:
$P^T = \{X_1^T, ..., X_N^T\}, X_i^T = (x_i^T, y_i^T)$
8: Calcular o modelo de aparência do *superpixel*:
$A(X_i^T) = \mu_{X\,i}$
9: Definir a vizinhança do *superpixel*, representado pelo seu centróide:
$N(R_i)$
10: **end for**
11: **end for**
12: Selecionar os centróides a serem monitorados no primeiro quadro do vídeo
13: **for** 1 até n **do**
14: Monitorar o centróide analisando o custo de correspondência de um centróide no quadro T com todos os centróides selecionados do quadro $(T + 1)$:
$C(X_i^T, Y_j^{T+1})$
15: Armazenar a trajetória do centróide
16: **end for**
17: Retornar como saída o fluxo de trajetórias espaço-temporais

A seguir tem-se a descrição de cada linha do algoritmo.

Linha 1: Leitura do vídeo, extraindo uma sequência de imagens do vídeo.

Linhas 2 a 4: Cada imagem do vídeo é segmentada em *superpixel* utilizando o método definido em [25].

Linhas 5 a 11: Para cada quadro do vídeo e para cada *superpixel* do quadro é encontrado o centróide do *superpixel*, que é o ponto que irá representá-lo para desenhar as trajetórias. Para cada *superpixel*, também é calculado o modelo de aparência dos pixels que formam o *superpixel*. Além disso, é definida a vizinhança de cada *superpixel*, que é definida através da triangulação de Delaunay.

Linha 12: São escolhidos os *superpixels* que serão monitorados, esses *superpixels* devem pertencer à região de movimento no vídeo. As regiões de movimento são encontradas fazendo a subtração dos quadros do vídeo.

Linhas 13 a 16: Cada um dos pontos a ser monitorado deve ter sua trajetória desenhada utilizando todos os quadros. Para isso é analisado o custo de correspondência dos pontos quadro a quadro utilizando uma das equações de custo: 4.4, 4.7, 4.8. Para cada ponto haverá uma trajetória.

Linha 17: Retorna o fluxo de trajetórias do movimento, que é obtido pela junção das trajetórias de cada ponto monitorado.

Capítulo 5

Experimentos

Neste capítulo são apresentados os resultados de experimentos utilizando vídeos sintéticos e vídeos reais utilizando a metodologia proposta. O objetivo é mostrar que o método identifica trajetórias do movimento. Foram utilizados vídeos de diferentes bancos de dados e também vídeos, tanto sintéticos quanto reais, produzidos no decorrer do trabalho para testes específicos.

Para cada vídeo são extraídas as trajetórias do objeto em movimento. Para extrair essas trajetórias, cada vídeo passa por um processo de segmentação, no qual, todos os quadros do vídeo são segmentados em *superpixels*. A região do *superpixel* além de compreender pixels que possuem características de brilho e textura semelhantes, também são responsáveis por identificar quais regiões do objeto estão em movimento, uma vez que pixels pertencentes à região interna à borda do objeto não irão pertencer ao mesmo *superpixel* que compreende pixels fora do contorno do objeto.

As trajetórias são representadas por coordenadas tridimensionais, onde a primeira e segunda coordenadas são os pontos no espaço bidimensional de cada quadro e a terceira coordenada é a variação temporal dos quadros, indicando a ordem na sequência de imagens do vídeo, representando assim o quadro T do vídeo. O conjunto de trajetórias extraídas formam um fluxo representado por trajetórias das regiões em movimento.

Diversos fatores, além da qualidade do vídeo, podem interferir diretamente nos resultados. Neste trabalho são destacados dois fatores: a segmentação e os parâmetros que definem a relação de vizinhança e a ponderação dada à correspondência entre os *superpixels* e seus vizinhos. A segmentação do vídeo está diretamente relacionada com o desenho das trajetórias. Uma trajetória representativa é obtida se a segmentação for estável, isto é, *superpixels* correspondentes em quadros subsequentes devem ser segmentados de forma semelhante, ou seja, segmentação semelhante. Uma segmentação mal feita pode fazer com que trajetórias se percam ou ainda que duas trajetórias se unam, entre outros problemas. A relação de vizinhança influencia na obtenção das trajetórias devido ao fato que vizinhos de *superpixels* que serão conectados também devem ser conectados. Um exemplo dessa situação pode ser o seguinte: suponha ter o *superpixel* p no quadro T com vizinhos a e b

e que serão conectados ao *superpixel* q do quadro $(T+1)$ e aos seus vizinhos c e d. Ou seja, os vizinhos deverão ser conectados de forma que seja feita a melhor correspondência entre eles.

O peso dado à correspondência dos centróides e da vizinhança pode variar, como foi visto na Seção 4.3.3 do Capítulo 4. Essa variação é dada pelos parâmetros α e β, das Equações 4.4, 4.7 e 4.8, que ponderam, respectivamente, a correspondência dos centróides e a correspondência da vizinhança. Se for dado um peso muito grande à correspondência dos vizinhos e se a informação dos vizinhos não for significativa, a trajetória pode não representar bem o movimento, desviando-se do caminho seguido pelo movimento.

Para efetuar os experimentos foram realizados testes analisando a segmentação utilizada e a sensitividade dos parâmetros: número de vizinhos, α e β. Isso foi considerado como premissa para os experimentos, ou seja, antes de mostrar os experimentos, serão mostrados como chegou-se a conclusões de quais seriam os melhores parâmetros a serem utilizados a fim de obter melhores resultados. Sendo assim, o restante desse capitulo é organizado em duas partes: a primeira consta as premissas dos experimentos e na segunda parte os experimentos realizados. Para a representação de algumas trajetórias, além das trajetórias originais, foram colocadas trajetórias suavizadas utilizando interpolação. No entanto, optou-se por colocar em todos os exemplos as trajetórias originais.

A Seção 5.1, premissas dos experimentos, é subdividida em duas outras partes. A primeira mostra como a segmentação pode influenciar nos resultados. Para tal, serão mostrados dois experimentos, executados em vídeos sintéticos, relacionados à segmentação. O primeiro deles com uma bola de apenas uma cor se movimentando no decorrer do vídeo e o segundo com uma bola tricolor. A segunda parte das premissas dos experimentos diz respeito à análise dos parâmetros que definem a vizinhança, o peso dado à correspondência *superpixel* a *superpixel*, representado por α, e ao peso dado à correspondência da vizinhança, representado por β. Para analisar a quantidade de vizinhos a ser analisada foram realizados três experimentos, um com vídeo sintético e dois com vídeos reais modificando a quantidade de vizinhos sendo analisados. Para análise dos parâmetros α e β serão mostrados quatro testes, aplicados em dois vídeos reais, enfatizando a diferença ao colocar peso maior à correspondência da vizinhança e vice-versa.

Finalmente, na Seção 5.2, tem-se as trajetórias dos movimentos, onde são mostrados cinco experimentos. Os três primeiros, na seção de vídeos reais, mostram vídeos de movimentos humanos, especificamente do movimento andar visto por diferentes ângulos. Foram utilizados vídeos produzidos na Universidade Federal de Uberlândia, vídeos do banco de dados CAVIAR[1] e um vídeo do trabalho de Fossati *et al.* [12]. Na seção de vídeos sintéticos tem-se dois experimentos, um deles simulando o fluxo bi-direcional de pessoas andando, no qual foi utilizado um vídeo do trabalho de Pelechano [38] no projeto

[1]EC Funded CAVIAR project/IST 2001 37540, encontrado na URL: `http://homepages.inf.ed.ac.uk/rbf/CAVIAR/`

MACES. No segundo um vídeo com um cubo de Rubik, que se desloca em relação ao fundo das imagens do vídeo.

5.1 Premissas dos experimentos

Para ilustrar os fatores que podem interferir nos resultados, serão apresentados nesta Seção 6 experimentos. No primeiro experimento, Seção 5.1.1, será analisada a influência da segmentação, mostrando que para diferentes segmentações haverão diferentes resultados, que uma segmentação ruim terá como resultado uma trajetória não representativa. No segundo experimento, Seção 5.1.2, será analisada a sensitividade da vizinhança, onde será mostrado que a quantidade de vizinhos utilizada influencia nos resultados. E num terceiro experimento, Seção 5.1.2, será analisado a ponderação entre a influência do *superpixel* e de seus vizinhos na construção das trajetórias, isto é, será analisado o que deve influenciar mais na extração das trajetórias, se a correspondência centróide a centróide ou a correspondência entre vizinhos.

Além destes experimentos, também serão mostrados na Seção 5.2.1 testes de extração de trajetórias de movimentos humanos, utilizando a abordagem proposta.

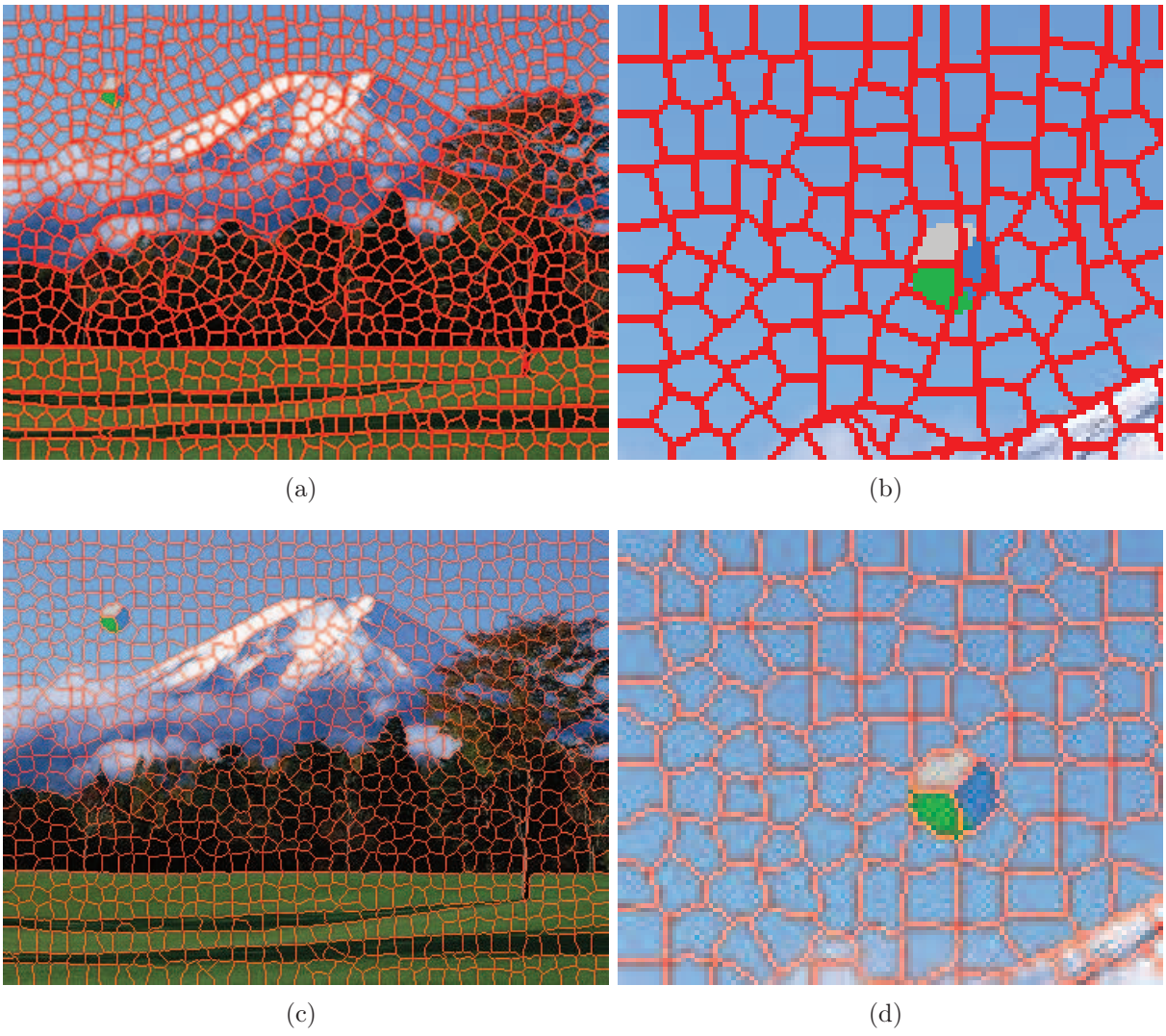
5.1.1 Segmentação

A proposta deste trabalho é segmentar quadros do vídeo em *superpixels*. Inicialmente foi utilizado a segmentação em *superpixels* de acordo com o trabalho de Mori [32], no entanto, os resultados obtidos não foram satisfatórios, ou seja, as trajetórias geradas não estavam condizentes com o movimento que estava sendo analisado. Numa análise bem mais detalhada do problema percebeu-se que no processo de segmentação, análogo ao dado em [32], ocorria a presença de padrões diferentes e de alta energia do contorno dentro de um mesmo *superpixel*, como por exemplo: partes de um objeto de uma mesma cor eram segmentadas em diferentes *superpixels*. Além disso, como existe, no método proposto em [32], um fator aleatório na inicialização dos *superpixels*, uma mesma imagem poderia gerar resultados diferentes em diferentes execuções do algoritmo. Os resultados obtidos na segmentação, como já mensionado no Capítulo 4, deixaram a desejar e desta forma, um outro algoritmo de segmentação em superperpixels, denominado *Turbopixels* [25], foi utilizado, trazendo melhores resultados.

Superpixels x *Turbopixels*

Para ilustrar as diferenças que podem ser obtidas utilizando as duas técnicas de segmentação, serão apresentados a seguir a aplicação dessas técnicas em um quadro de um vídeo de uma bola em movimento, num fundo com paisagem. A Figura 5.1 mostra em (a)

a segmentação utilizando *superpixels* de acordo com a proposta de [32] e em (c) a segmentação utilizando o algoritmo *Turbopixels* [25]. Como pode-se notar, em (b), a primeira segmentação não delimita exatamente as regiões de acordo com sua cor, trazendo cores diferentes em uma mesma região, como visto nas partes da bola. A segunda proposta de segmentação em *superpixels* contempla melhor esta questão (Figuras 5.1 (c) e (d)).

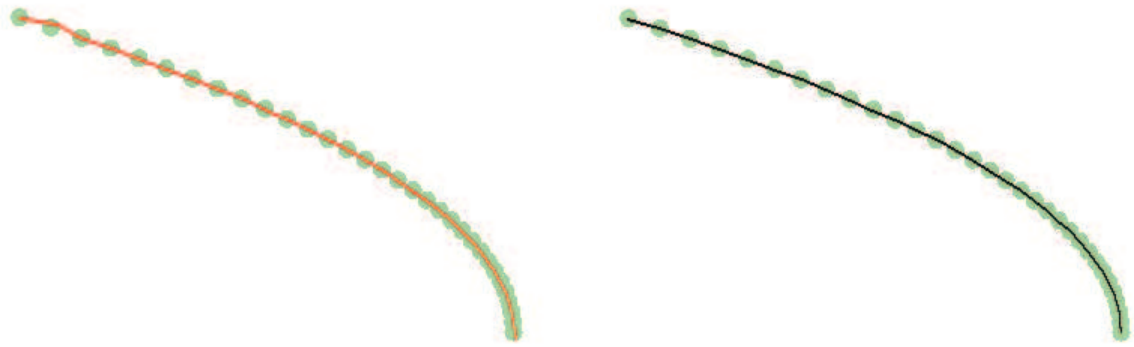


(a) (b)

(c) (d)

Figura 5.1: (a) Imagem segmentada em *superpixels* de acordo com o trabalho de [32]; (b) zoom de (a); (c) Imagem segmentada em *superpixels*, utilizando o algoritmo *Turbopixels* [25]; (d) zoom de (c).

Para vídeos simples, compostos por imagens com poucas cores a diferença na segmentação não traz grandes impactos na extração das trajetórias, como pode ser observado na Figura 5.2, onde é exibida a trajetória de uma bola verde em um fundo branco. A Figura 5.2 (a) mostra a trajetória extraída com a segmentação em *superpixels* e em (b) a partir da segmentação utilizando *Turbopixels*. Como pode ser visualizado as trajetórias obtidas são praticamente as mesmas. Nessa figura todos os quadros do vídeo foram reunidos em apenas uma imagem para mostrar o movimento da bola.

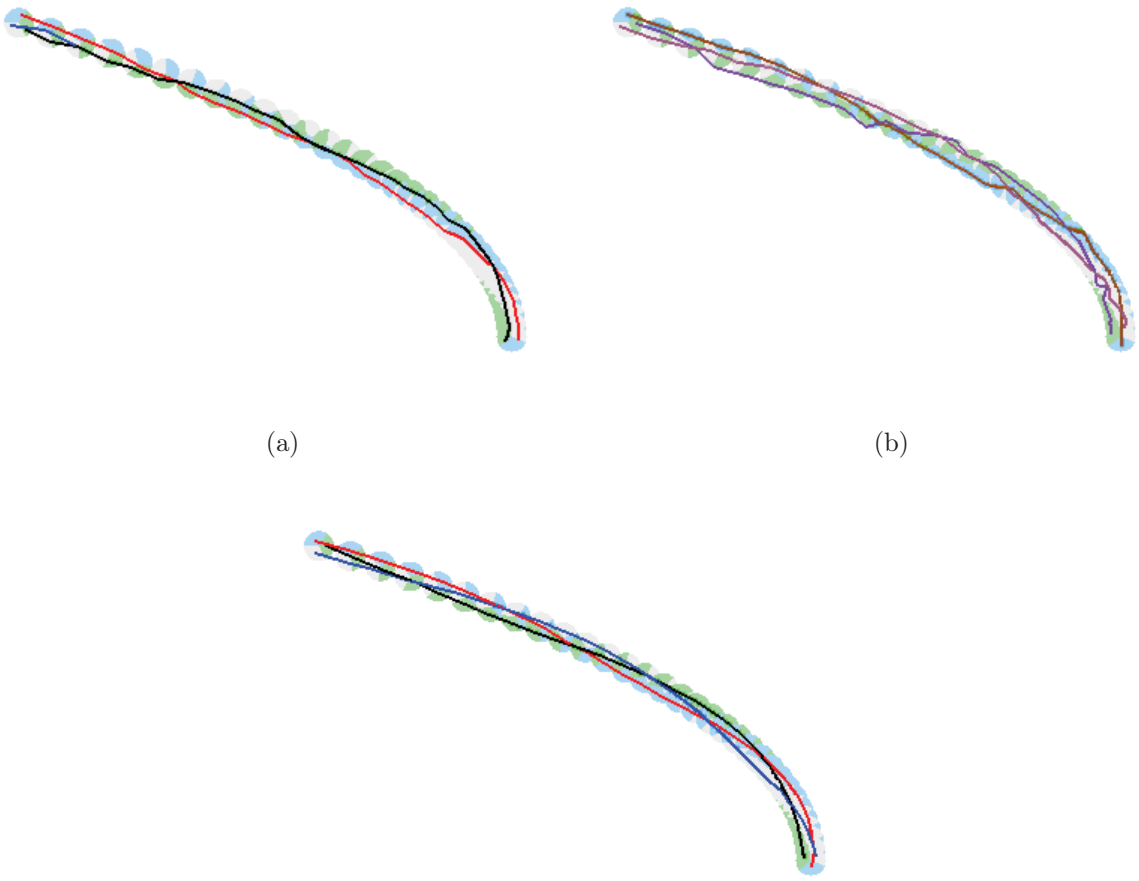


(a) (b)

Figura 5.2: Trajetória de uma bola verde em um fundo branco. (a) Utilizando segmentação em *superpixels*; (b) Utilizando o algoritmo *Turbopixels* para segmentação em *superpixels*.

No entanto, em vídeos com um pouco mais de detalhes, como por exemplo, uma bola tricolor em movimento, há uma diferença notável nas trajetórias extraídas. A Figura 5.3 exibe o resultado da representação de trajetórias dos *superpixels* que compõem a bola utilizando os dois métodos de segmentação. Em (a) utilizando o método de segmentação em *superpixels* usado em [32] e em (b) utilizando *Turbopixels*. Nessa figura é possível perceber que duas trajetórias, que estão nas cores azul e preta, se unem, ou seja, em determinado ponto elas atingem o mesmo *superpixel* e daí em diante seguem o mesmo caminho quando a segmentação usada for a dada em [32]. Essa situação não acontece ao utilizar o algoritmo *Turbopixels*. As trajetórias, mostradas na Figura 5.3 (b), seguem o movimento da bola, que além de se movimentar em relação ao fundo, também gira em seu próprio eixo.

Para os testes apresentados nesta seção, os quadros que compõem o vídeo foram segmentados em aproximadamente 1000 *superpixels* cada. Essa quantidade de *superpixels* em cada quadro pode variar de acordo com o tipo de resolução do vídeo. A quantidade de *superpixels* em um vídeo com resolução 300 $\times$ 400 é menor que em um vídeo com 450 $\times$ 600, por exemplo.



(a) (b)

(c)

Figura 5.3: Trajetória de uma bola tricolor em um fundo branco. (a) Utilizando segmentação em *superpixels* dada em [32]; (b) Utilizando o algoritmo *Turbopixels* [25] para segmentação em *superpixels*; (c) Versão suavizada de (b) utilizando *Spline*.

5.1.2 Sensitividade dos Parâmetros

Para desenhar o fluxo de trajetórias, utilizando a proposta deste trabalho, é necessário analisar o custo de correspondência entre os *superpixels* em quadros subsequentes. O custo (Equações 4.4, 4.7 e 4.8) é dado ponderando a correspondência entre os *superpixels*, por meio da diferença entre os valores dados pelo modelo de aparência e ponderando o custo de correspondência da vizinhança dos dois *superpixels* que estão sendo analisados.

Dessa forma, é necessário definir quais são os melhores valores para os parâmetros α, β e quantidade de vizinhos, de acordo com as Equações 4.4, 4.7 e 4.8, para minimizar o custo de correspondência, obtendo trajetórias representativas ao fazer a ligação entre os *superpixels* de maior concordância de características.

Nas seções seguintes serão mostrados experimentos relativos aos parâmetros desta equação, analisando quais os impactos ao variá-los. Primeiramente será analisada a sensitividade da vizinhança e depois dos parâmetros α e β.

Análise da Sensitividade da Vizinhança

Para definir a relação de vizinhança entre os centróides foi utilizada a triangulação de Delaunay, na qual, dois *superpixels*, representados por seus centróides, são vizinhos se existir uma aresta que os conecta. O número de vizinhos de cada centróide num mesmo quadro pode ser diferente, dependendo das características de textura, energia do contorno, entre outras, ao redor do *superpixel*.

De posse da relação de vizinhança, é necessário calcular o custo de correspondência da vizinhança dos dois *superpixels*, que é dado pelo somatório das diferenças dos valores dados pelo modelo de aparência dos vizinhos correspondentes. Como a quantidade de vizinhos pode variar, torna-se necessário definir qual a melhor quantidade de vizinhos que deve ser analisada para obter resultados mais eficientes.

Vários testes foram realizados variando a quantidade de vizinhos. O número oito foi a quantidade máxima de vizinhos considerada. Para cada vídeo foram testadas todas as possibilidades de correspondências de vizinhos. Os resultados mostraram que ao analisar os vizinhos é melhor que a quantidade de vizinhos seja maior. Ou seja, quanto maior for a quantidade de vizinhos, mais precisas serão as trajetórias. Nas Figuras 5.4 e 5.5 pode-se observar esse fato, onde tem-se, respectivamente, trajetórias identificadas com a análise de apenas dois vizinhos e com a análise de todos os vizinhos dos *superpixels*. Com poucos vizinhos a trajetória se perde de tal maneira que prossegue para um caminho diferente daquele que segue o movimento.

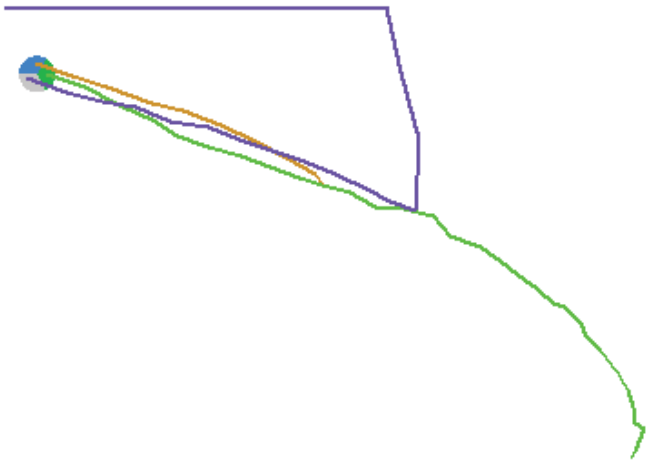


Figura 5.4: Trajetórias considerando o peso de correspondência de dois vizinhos para cada *superpixel*.

Outra situação em que utilizar o peso de correspondência dos vizinhos auxilia é na consistência das trajetórias. Ou seja, quando nenhum vizinho é considerado para a análise de continuidade das trajetórias, elas podem unir-se, pois chegam a um mesmo *superpixel* e seguem a partir desse momento somente um caminho. Essa situação pode ocorrer quando

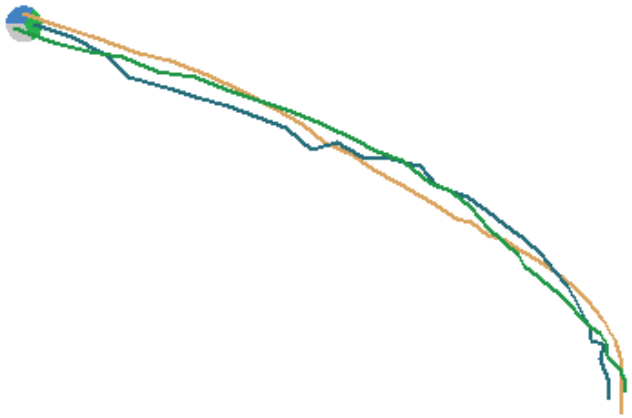
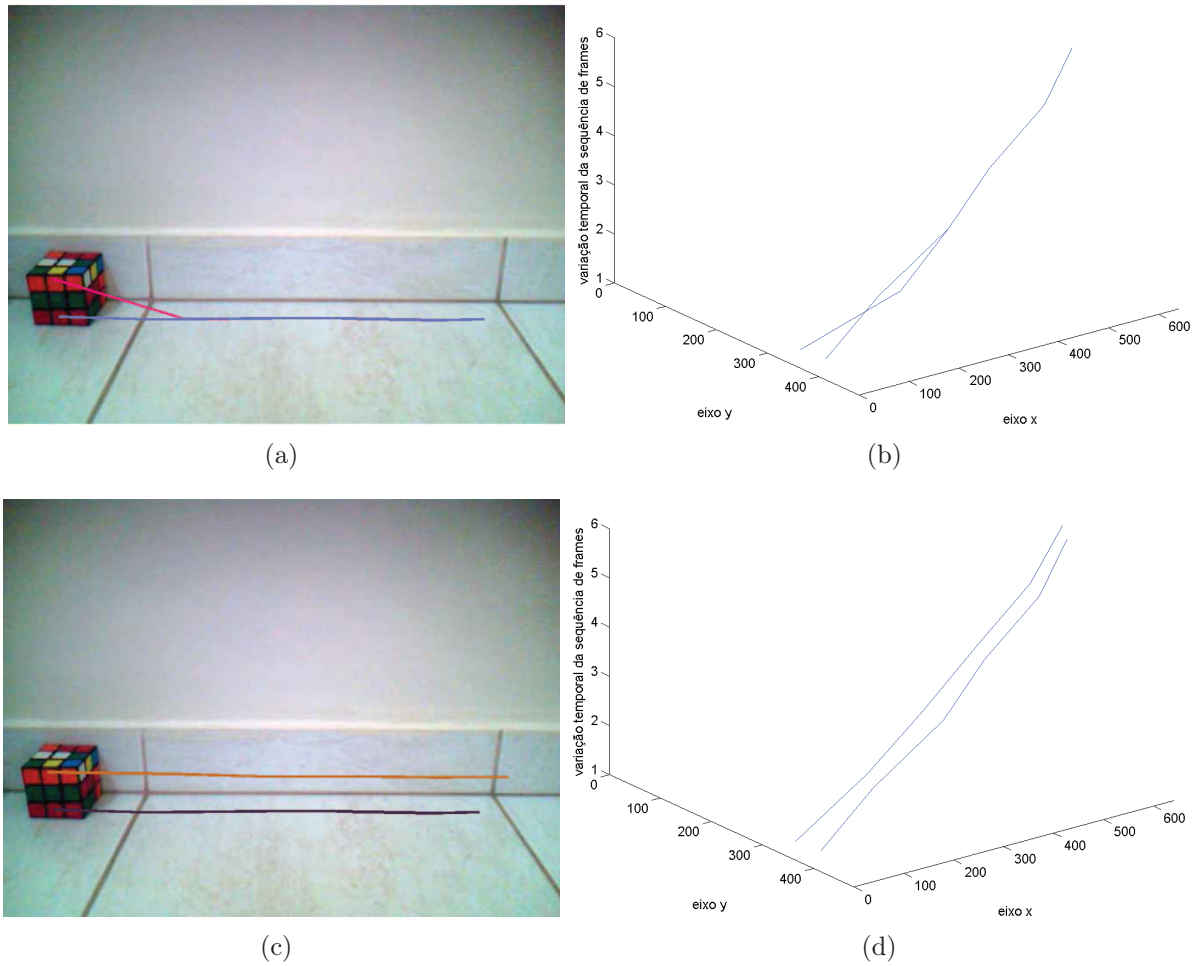


Figura 5.5: Trajetórias considerando o peso de correspondência de oito vizinhos para cada *superpixel*.

tem-se dois *superpixels* com aparências semelhantes e não há critério para decisão, não sendo possível definir qual será o centróide que dará continuidade à trajetória. Nesse caso, dois centróides do quadro T têm continuidade em um mesmo centróide no quadro $(T+1)$ e a partir deste momento as duas trajetórias se unem numa única. A correspondência da vizinhança pode ser interpretada como um critério de desempate quando essa situação ocorre. Dessa forma, além de avaliar se os *superpixels* têm aparências semelhantes, também é analisada se sua vizinhança permanece a mesma. As Figuras 5.6 (a)-(d) exemplificam este cenário. Nas Figuras 5.6 (a) e (b) tem-se um experimento com um vídeo, onde o objeto em movimento é um cubo de Rubik, popularmente conhecido como "cubo mágico". Neste exemplo não foram utilizados vizinhos para fazer a correspondência entre os centróides. O resultado foi a união de duas trajetórias, que se uniram devido a característica dos *superpixels* serem semelhantes. No entanto, ao analisar a vizinhança, as trajetórias que antes se uniam agora seguem seus caminhos corretos, como pode ser observado nas Figuras 5.6 (c) e (d).

O próximo exemplo ilustra a diferença de resultados considerando diferentes números de vizinhos. Na Figura 5.7 é mostrado um exemplo de vídeo com movimento humano, no qual o ator, de pé movimenta seus braços para cima e para baixo formando um círculo ao redor de seus ombros e cabeça. As Figuras 5.7 (a) e (b) mostram o resultado obtido utilizando somente dois vizinhos. Ao analisar apenas dois vizinhos as trajetórias ficam completamente inconsistentes, não sendo possível interpretar qual é o movimento que está sendo realizado. Já nas Figuras 5.7 (c) e (d) é mostrado o resultado ao utilizar todos os vizinhos. Ao analisar toda a vizinhança do centróide, o resultado melhora consideravelmente, as trajetórias das mãos e braços conseguem seguir o movimento realizado. Vale lembrar que para a análise da vizinhança são utilizados os mesmos valores para α e β, modificando apenas a quantidade de vizinhos.



(a) (b)

(c) (d)

Figura 5.6: Diferença das trajetórias ao utilizar ou não a vizinhança dos *superpixels*. (a) e (b) Trajetórias extraídas sem analisar a vizinhança; (c) (d) Trajetórias extraídas considerando a vizinhança dos *superpixels*.

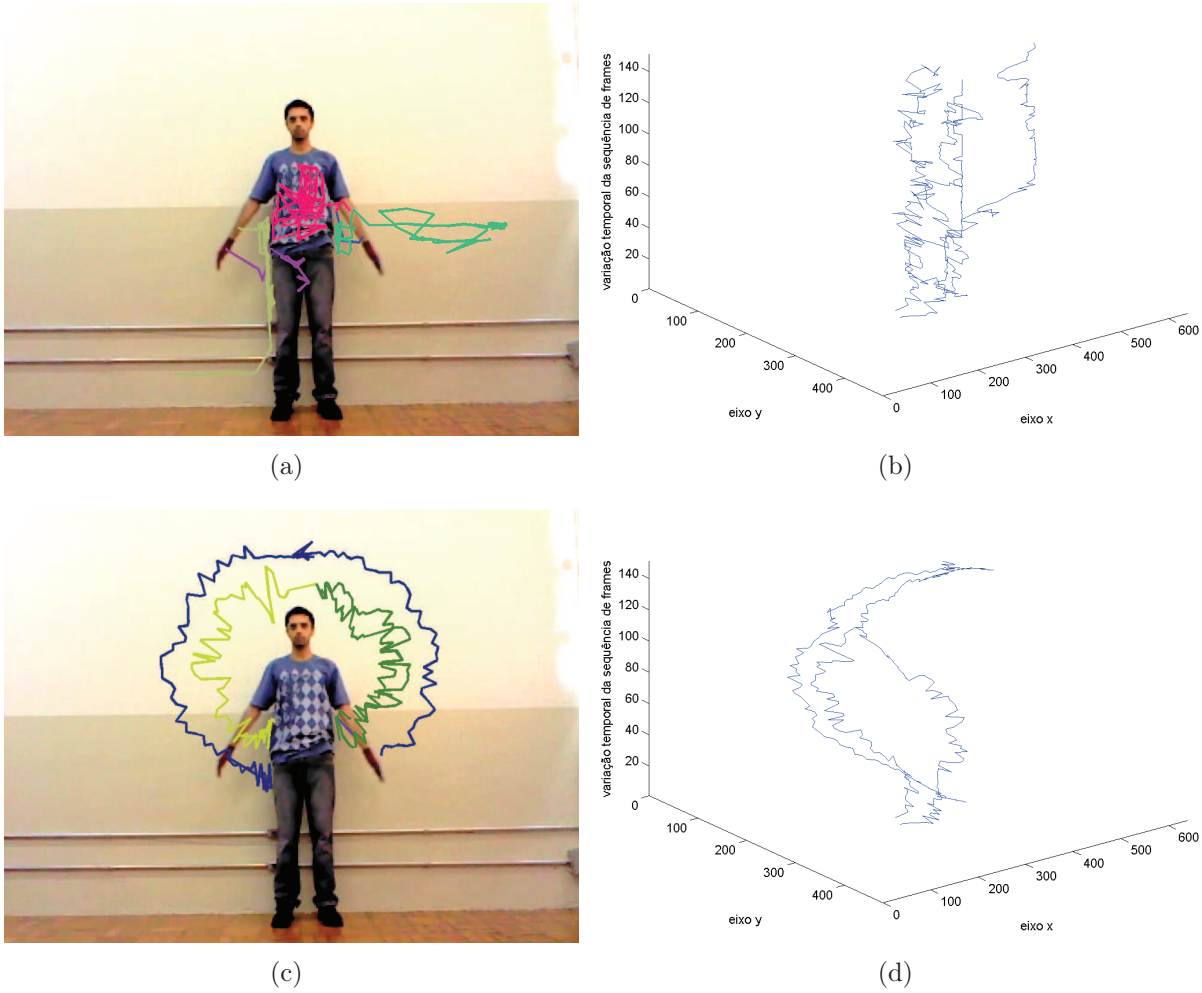


Figura 5.7: Movimento "círculo ao redor da cabeça e ombros". Diferença das trajetórias ao utilizar apenas dois vizinhos (a) e (b) e ao utilizar toda a vizinhança (c) e (d) dos *superpixels*.

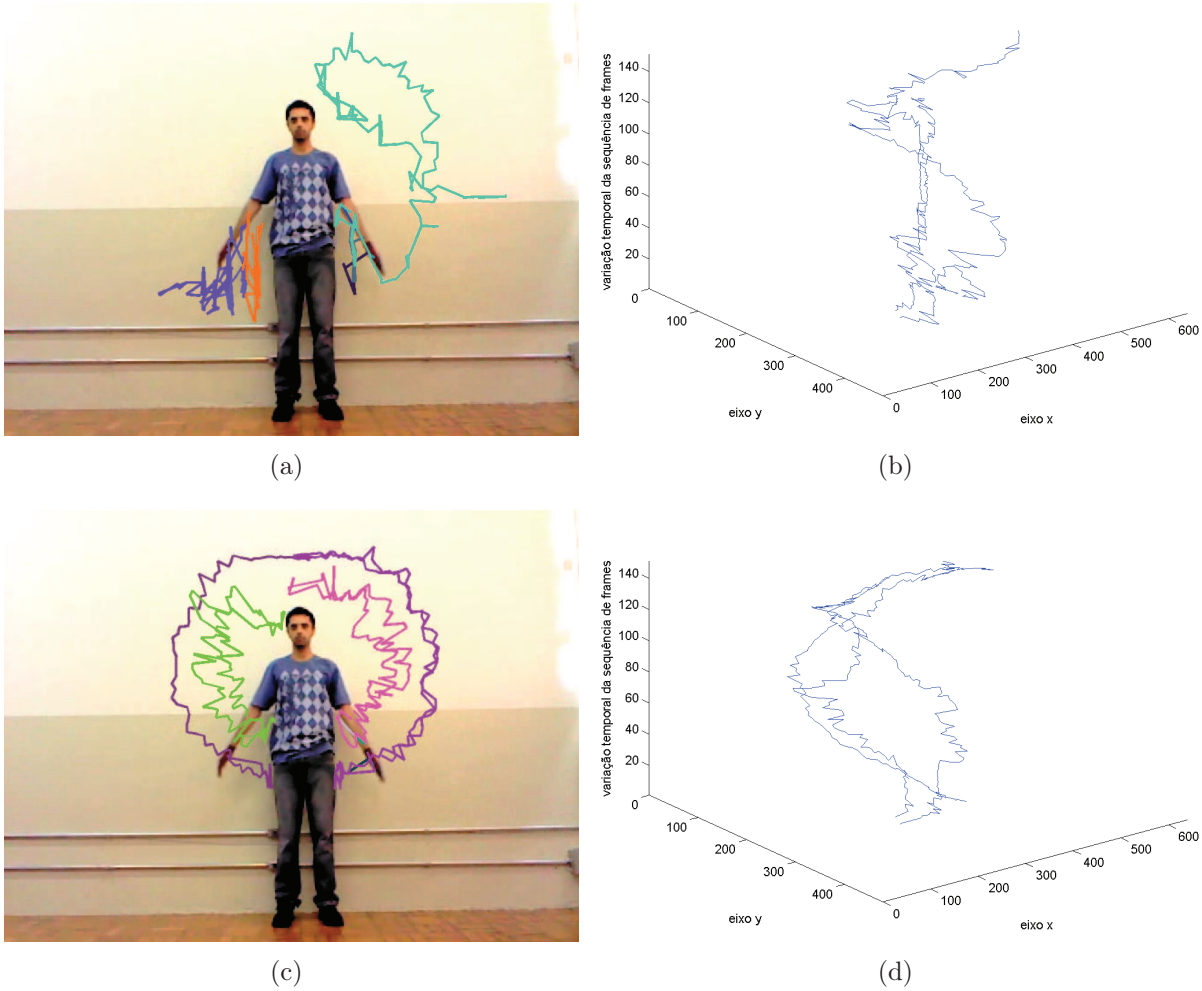
Análise da Sensitividade para os Parâmetros α e β

Além de ter uma boa correspondência da vizinhança é necessário ponderar essa correspondência e ponderar também a correspondência centróide a centróide ao longo das trajetórias. Esses pesos são representados pelos parâmetros α e β, respectivamente, definidos nestes testes pela Equação de custo 4.8:

$$C(X_i^T, Y_j^{T+1}) = \alpha\,(\delta(X_i^T, Y_j^{T+1})) + \beta\ \ (\delta(\frac{1}{|N(X_i)|}\Sigma_{(X_{p_i}\epsilon N(X_i))}X_{p_i}, \frac{1}{|N(Y_j)|}\Sigma_{(Y_{p_j}\epsilon N(Y_j))}Y_{p_j}))$$

Os parâmetros α e β utilizados foram definidos de forma que $\alpha+\beta = 1$. De acordo com os resultados dos testes realizados, concluiu-se que $\alpha \geq \beta$. Na Figura 5.8 são mostrados exemplos utilizando variações nos valores para α e β. Na Figura 5.8 (a) e (b) tem-se $\alpha = \frac{1}{6}$ e $\beta = \frac{5}{6}$, o que não traz bons resultados, enquanto que em (c) e (d), onde tem-se $\alpha = \frac{4}{5}$ e $\beta = \frac{1}{5}$, tem-se resultados bem melhores.

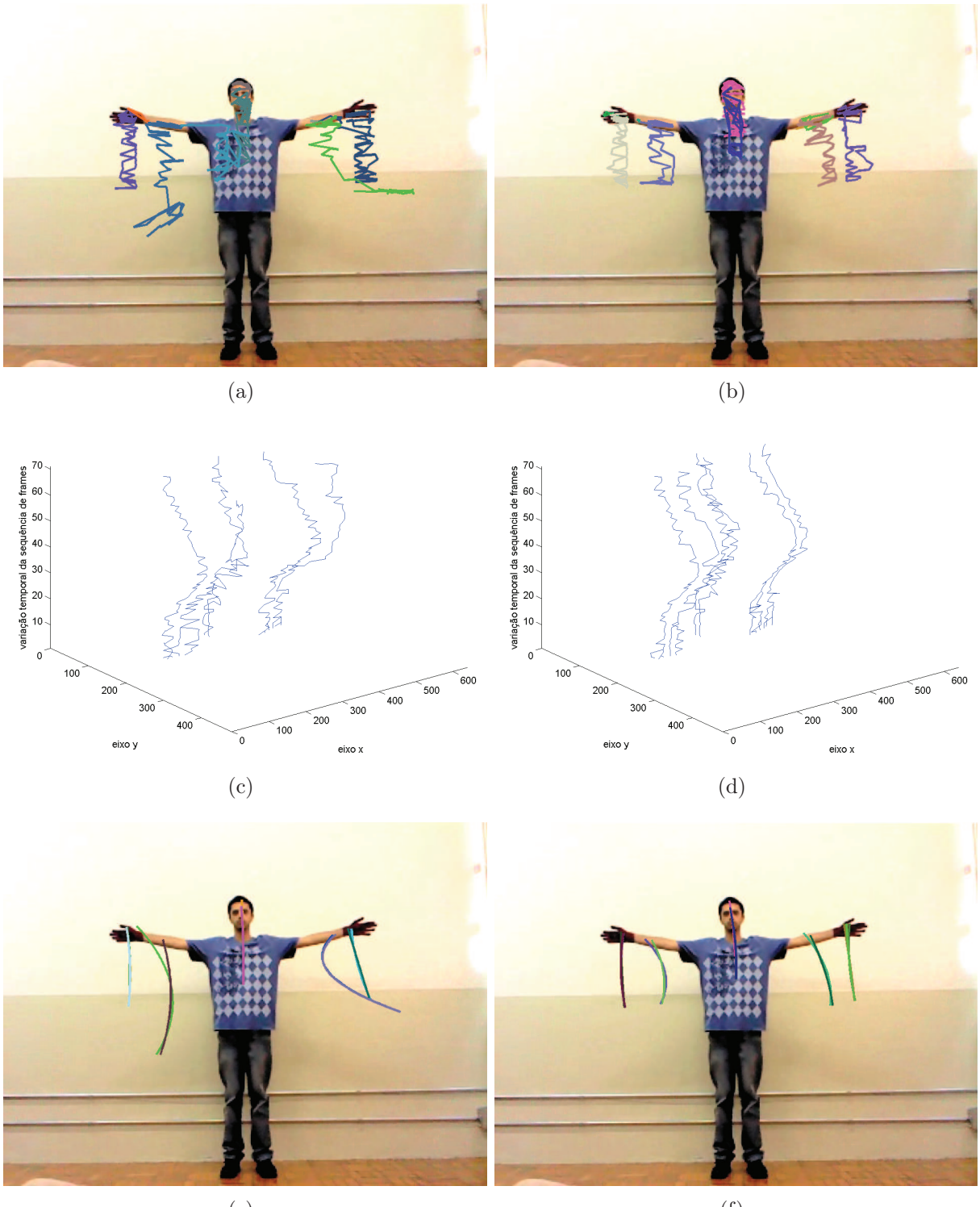
Outro exemplo para a relação entre α e β pode ser visto na Figura 5.9. Nesse vídeo o ator levanta-se e abaixa-se com os braços estendidos. Na Figura 5.9 (a), (c) e (e), com $\alpha = \frac{1}{4}$ e $\beta = \frac{3}{4}$, os resultados obtidos usando $\alpha < \beta$ não foram relevantes e em (b), (d) e (f), com $\alpha = \frac{5}{7}$ e $\beta = \frac{2}{7}$, foram obtidos resultados melhores.



(a) (b)

(c) (d)

Figura 5.8: Movimento "círculo ao redor da cabeça e ombros". Diferença das trajetórias ao utilizar toda a vizinhança dos *superpixels* e variando os parâmetros α e β. (a) e (b) $\alpha = \frac{1}{6}$ e $\beta = \frac{5}{6}$; (c) e (d) $\alpha = \frac{4}{5}$ e $\beta = \frac{1}{5}$.



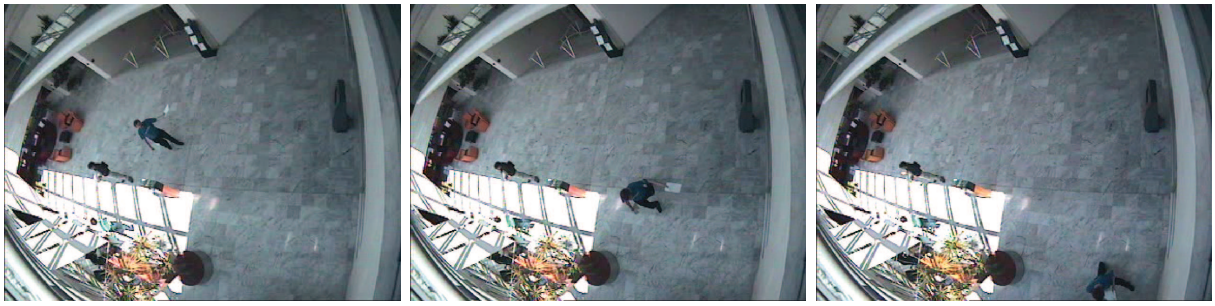
(a) (b) (c) (d) (e) (f)

Figura 5.9: Movimento "abaixar e levantar-se com os braços estendidos". Diferença das trajetórias ao utilizar toda a vizinhança dos *superpixels* e variando os parâmetros α e β. (a), (c) e (e) $\alpha = \frac{1}{4}$ e $\beta = \frac{3}{4}$; (b), (d) e (f) $\alpha = \frac{5}{7}$ e $\beta = \frac{2}{7}$. (e) e (f) são as suavizações de (a) e (b), respectivamente, utilizando interpolação com o Método dos Mínimos Quadrados.

5.2 Trajetórias de Movimentos

5.2.1 Experimentos com Vídeos Reais

O primeiro exemplo de extração de trajetórias do movimento a ser ilustrado é o de um vídeo da base de dados CAVIAR[2]. O vídeo tem resolução de 384 × 288 pixels a uma taxa de 25 quadros por segundo, totalizando 125 quadros. Nesse vídeo tem-se a visão do movimento por meio de uma câmera angular localizada no canto da cena. O trecho do vídeo analisado mostra um homem andando. Inicialmente é possível ver todo o corpo do homem, mas no decorrer do vídeo partes do corpo ficam oclusas devido a posição da câmera. Na Figura 5.10 tem-se três quadros do vídeo: o primeiro, um quadro intermediário e o último quadro.



(a) (b) (c)

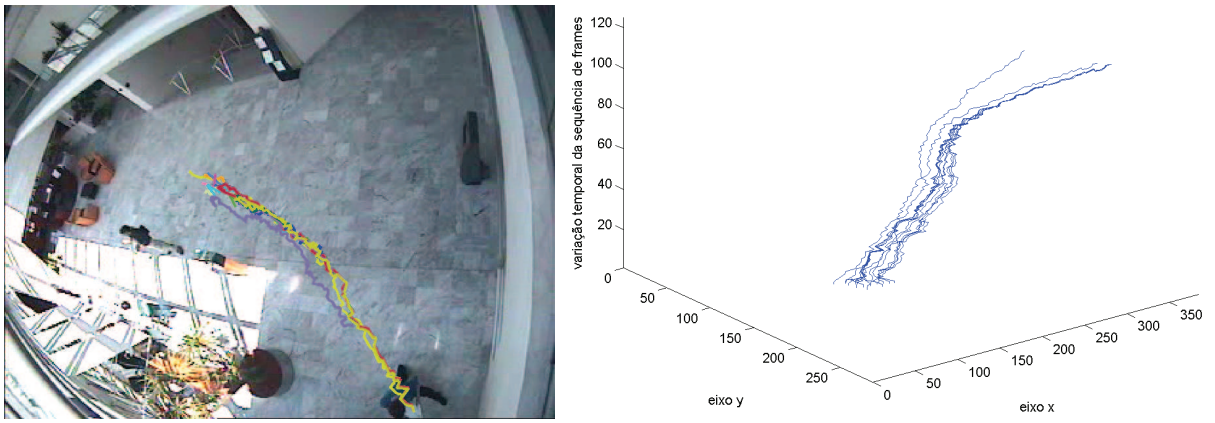
Figura 5.10: Sequência com três quadros do movimento "andar", primeiro experimento. (a) Primeiro quadro; (b) Quadro intermediário e (c) Último quadro.

A representação das trajetórias para esse movimento é mostrada na Figura 5.11, onde nota-se que no início do movimento, quando é possível ver todo o corpo do ator, tem-se várias trajetórias e no final do movimento, tem-se apenas algumas trajetórias, que são aquelas que monitoram partes do objeto que não ficaram oclusas no decorrer do vídeo.

O segundo experimento foi realizado em um vídeo, do trabalho de Fossati *et al* [12], com resolução de 360 × 288 pixels, taxa de 25 quadros por segundo e 59 quadros. Neste vídeo, o movimento também é de uma pessoa andando, onde a visão da câmera permite observar o ator inicialmente de frente e ao longo do vídeo de costas. Na Figura 5.12 são mostrados três quadros desse vídeo. Na Figura 5.13 pode-se observar as trajetórias do movimento "andar" geradas.

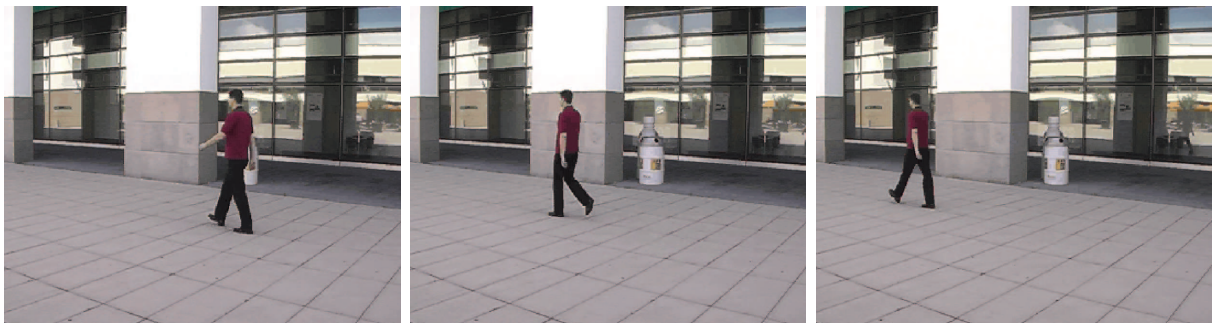
O terceiro experimento foi realizado com um vídeo de resolução 640 × 480 a uma taxa de 25 quadros por segundo e 95 quadros, gravado experimentalmente na Universidade Federal de Uberlândia. O vídeo mostra o movimento de uma pessoa andando, no qual a câmera encontra-se parada. Os quadros 1, 43 e 95 são mostrados na Figura 5.14 para dar a noção do movimento realizado.

[2]EC Funded CAVIAR project/IST 2001 37540, encontrado na URL: `http://homepages.inf.ed.ac.uk/rbf/CAVIAR/`



(a) (b)

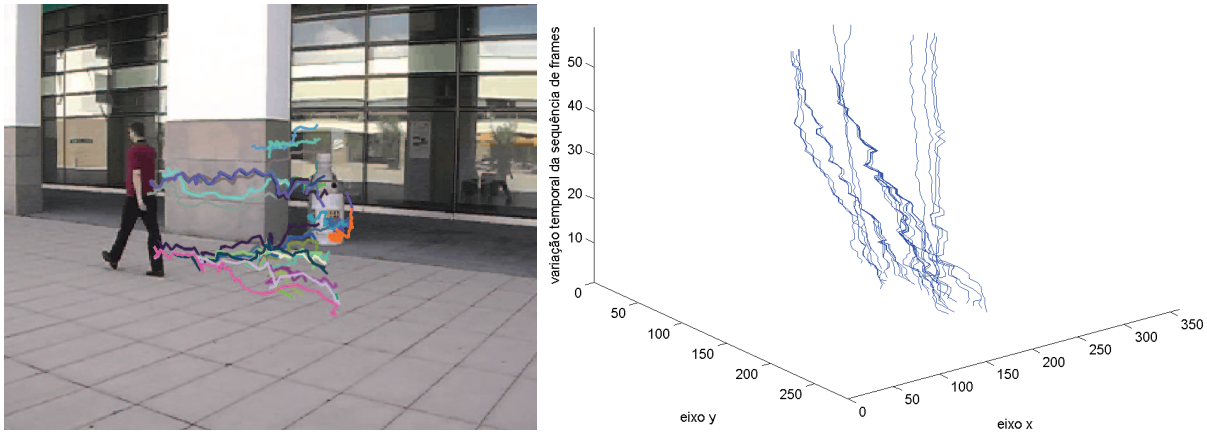
Figura 5.11: Trajetórias do movimento “andar”.



(a) (b) (c)

Figura 5.12: Sequência com três quadros do movimento andar, segundo experimento. (a) Quadro 1; (b) Quadro 30 e (c) Quadro 59.

O resultado do monitoramento do movimento, nesse vídeo, pode ser acompanhado na Figura 5.15. As trajetórias à partir das regiões da cabeça do ator descrevem o caminho que o ator percorre. As trajetórias que se iniciam no braço esquerdo do ator se unem em determinado ponto do vídeo, isso ocorre porque os vizinhos e os valores dados pelo modelo de aparência para os *superpixels* do braço são semelhantes. Essas trajetórias conseguem representar o movimento do início ao fim do vídeo, o que não acontece com a trajetória da mão direita do ator, a qual em certo ponto do vídeo se desvia do trajeto correto, pois a mão fica escondida atrás das pernas do ator. Nesse momento a trajetória tem continuação em *superpixels* na calça do ator, que tem característica mais semelhante, daí em diante começa a seguir por um caminho que mais se assemelha aos *superpixels* da calça. Isto pode ser evitado se para a continuidade do movimento no quadro $(T+1)$ forem analisados os quadros $(T-l)$; $l = 0, ..., k$; $k < T$, analogamente como o quadro T é analisado. A exploração desta proposta será realizada em trabalhos futuros.

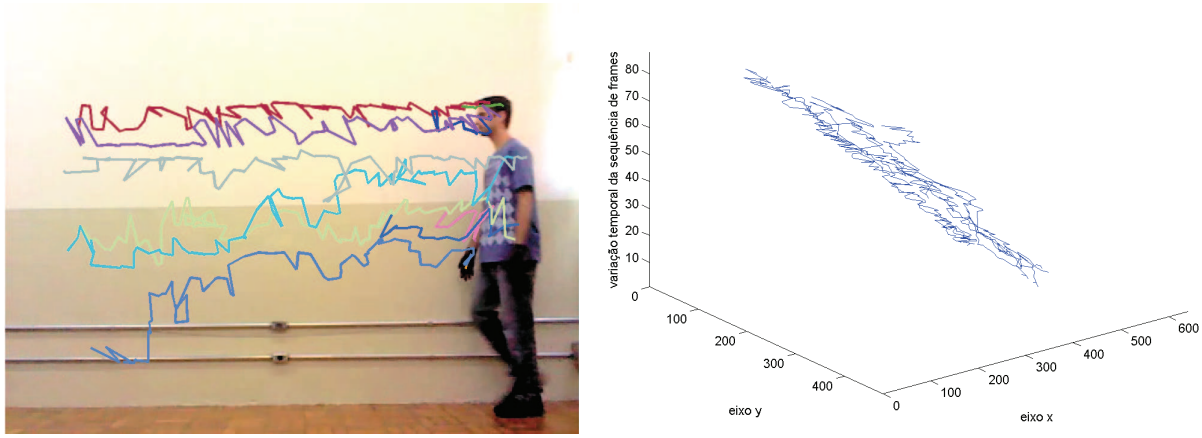


(a) (b)

Figura 5.13: Trajetórias do movimento “andar”.

(a) (b) (c)

Figura 5.14: Sequência com três quadros do movimento andar. (a) Quadro 1; (b) Quadro 43 e (c) Quadro 95.



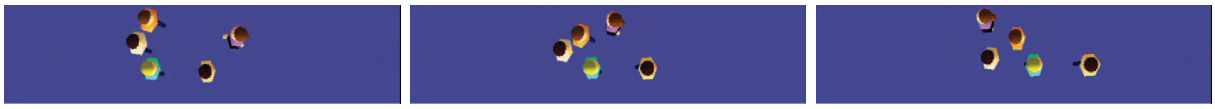
(a) (b)

Figura 5.15: Trajetórias do movimento “andar”.

5.2.2 Experimentos com Vídeos Sintéticos

O primeiro experimento com vídeos sintéticos visa mostrar as trajetórias do movimento de várias pessoas caminhando em sentido bi-direcional. O vídeo analisado é um dos vídeos

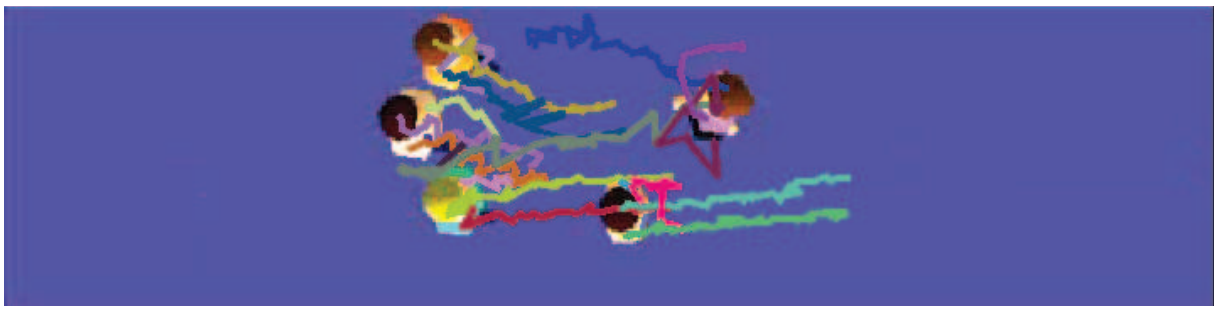
do projeto MACES[3], um sistema de simulação de multidão para análise de comportamento [38]. Esse vídeo, com 60 quadros, simula o comportamento de cinco "pessoas" que se deslocam em direções contrárias. A Figura 5.16 exibe três quadros do vídeo, mostrando o sentido de cada uma das pessoas. As quatro pessoas mais a esquerda no vídeo se deslocam da esquerda para a direita e a pessoa mais a direita se desloca da direita para a esquerda.



(a) (b) (c)

Figura 5.16: Quadros de um vídeo do projeto MACES.

O resultado do monitoramento é mostrado na Figura 5.17, na qual é possível observar que as trajetórias representam o caminho percorrido pelas pessoas indicando o fluxo do movimento.



(a)

Figura 5.17: Trajetórias do movimento "andar" com vários objetos vistos de cima.

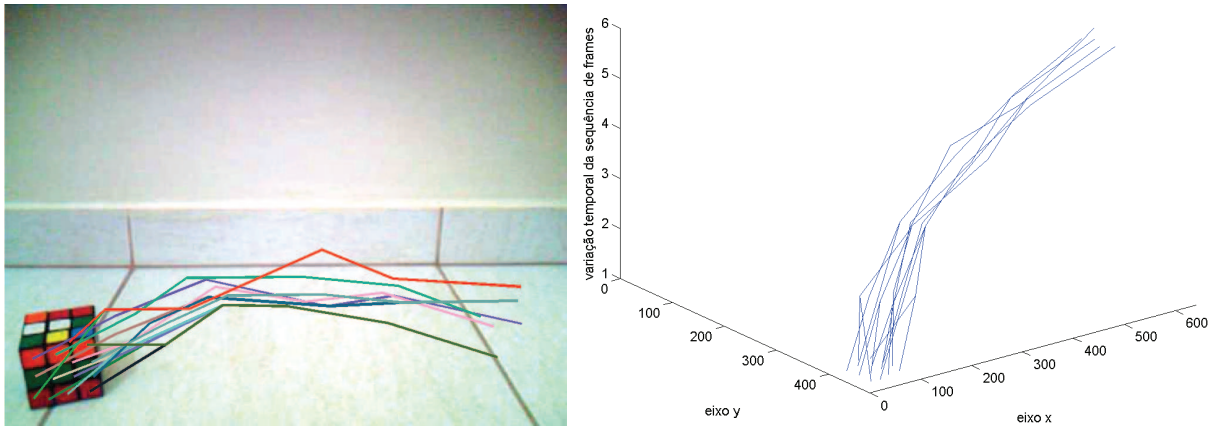
O segundo experimento com vídeos sintéticos faz a extração das trajetórias das regiões de uma das faces de um cubo que se desloca no decorrer da sequência de quadros do vídeo. Esse experimento foi realizado para observar o comportamento do método ao utilizar diferentes cores próximas umas das outras, tornando a vizinhança bem heterogênea. A Figura 5.18 exibe a sequência de quadros utilizada para montar as trajetórias do movimento de um cubo.

(a) (b) (c) (d) (e)

Figura 5.18: Quadros do vídeo cubo de Rubik.

[3]Pode ser acessado em `http://cg.cis.upenn.edu/hms/people/pelechano/MACES/`

O resultado desse experimento pode ser observado na Figura 5.19, onde é possível analisar as trajetórias do movimento das regiões de uma das faces do cubo. Pode-se observar que algumas trajetórias se unem no início do vídeo, isso ocorre porque há confusão quanto aos vizinhos, já que existe uma grande região de cor preta ao redor dos quadrados coloridos, que se unem aos quadrados coloridos na segmentação em *superpixels*. Este é o grande problema causado devido a baixa resolução dos vídeos, pois as cores se misturam nas regiões de contorno para suavizar a imagem. Ainda assim, foi possível extrair as trajetórias do movimento de forma a perceber o fluxo do movimento.



(a) (b)

Figura 5.19: Trajetórias do movimento do vídeo cubo de Rubik.

5.3 Análise do Método

- Com os resultados mostrados nas seções anteriores nota-se a eficácia do método proposto por este trabalho. Os resultados poderiam ser melhores se não fossem os problemas relativos à resolução das imagens do vídeo e da segmentação em *superpixels*, que mesmo utilizando o algoritmo *Turbopixels*, ainda apresenta alguns problemas.
- Mesmo sem utilizar uma segmentação ideal, os resultados foram bastante satisfatórios. Considerando a complexidade do problema, as trajetórias extraídas foram consistentes com os movimentos dos objetos.
- Utilizando-se a vizinhança, as trajetórias ficam bem definidas, ou seja, as trajetórias não tem continuidade somente porque os centróides têm o mesmo valor de aparência, mas também porque os vizinhos apresentam similaridade na aparência. Quando existir mais de um centróide com a mesma característica, a escolha de continuidade da trajetória, é dada pela vizinhança do ponto. Trajetórias construídas sem a análise dos vizinhos podem ficar esparsas, pois várias trajetórias podem se unir num mesmo ponto e a partir daí seguirem o mesmo caminho.

- Quanto maior a quantidade de vizinhos melhores serão os resultados, porque os vizinhos podem rotacionar ao redor do *superpixel* dependendo do tipo de movimento. Dessa forma, utilizar a quantidade máxima de vizinhos, aumenta a probabilidade de que sejam os mesmos e a correspondência entre os vizinhos seja ótima, ou seja, que os vizinhos se encaixem adequadamente.

Capítulo 6

Conclusões e Trabalhos Futuros

Neste capítulo, são apresentadas conclusões obtidas à partir dos experimentos realizados ao longo desta dissertação, no que diz respeito ao monitoramento de movimento em vídeo. Ao final do capítulo, são apresentados possíveis novos experimentos e linhas de investigação futuras.

6.1 Conclusões

Este trabalho descreveu um método de representação do movimento, por meio de trajetórias, baseadando-se nas premissas do método *Iterated Conditional Modes*(ICM). Os parâmetros de aparência local do *superpixel* e a consistência da vizinhança foram estimados, via ICM, baseando-se nas premissas de que pixels que pertencem a um objeto em movimento não sofrerão mudanças abruptas em sua cor ou brilho através do vídeo e que as características da vizinhança de um pixel devem permanecer similares em quadros consecutivos.

Na abordagem proposta foram utilizados pontos dentro do contorno do objeto para monitorar o seu movimento. Esses pontos foram dados pelos centróides dos *superpixels* que compõem o objeto. Para realizar o monitoramento, montando as trajetórias, foi utilizada uma função de custo baseada nas premissas descritas anteriormente. Com o resultado da função custo aplicada a todos os *superpixels* possíveis de monitoramento, tem-se o *superpixel* que deve dar continuidade à trajetória, no caso desse *superpixel* ter o menor custo de correspondência.

Foram realizados testes utilizando vídeos sintéticos e vídeos reais para monitoramento do movimento "andar". Os testes envolveram análise dos parâmetros α e β, para ponderar, respectivamente, o custo de correspondência dos *superpixels* e o custo de correspondência da vizinhança dos *superpixels*, análise da quantidade de vizinhos a ser utilizada e o método de segmentação em *superpixels*. De acordo com os testes realizados foi constatado que deve-se utilizar toda a vizinhança dos *superpixels* para realizar o monitoramento. Além disso, o peso de correspondência entre os *superpixels* deve ser maior que o peso de cor-

respondência da vizinhança. Outro fator importante para obter trajetórias consistentes é ter uma boa segmentação das imagens que compõem o vídeo.

Os resultados dos testes realizados utilizando o método proposto foram bastante animadores. Na maioria dos movimentos em vídeos monitorados as trajetórias foram representativas.

Um dos problemas encontrados para validação do método proposto foi a baixa resolução dos vídeos, que impacta sobre a qualidade de segmentação em *superpixels*, ou seja, boa qualidade dos *superpixels* depende da qualidade do vídeo: resolução e quadros por segundo. Outro problema encontrado ao realizar os experimentos foi o tempo de resposta. Vídeos com boa resolução apresentam problemas com processamento computacional.

6.2 Trabalhos Futuros e Aplicações

A proposta deste trabalho permite seguir por vários caminhos para melhorar os resultados e também permite aplicações do mesmo em diversas áreas. A seguir tem-se uma breve descrição do que pode ser trabalhado utilizando esta proposta e algumas das possíveis aplicações deste trabalho, melhorando a interação homem-máquina. Seguem sugestões de trabalhos futuros e aplicações:

Vertentes para futuros trabalhos:

- Para melhorar a consistência das trajetórias pode-se analisar o histórico das mesmas, avaliando o modelo de aparência de pontos anteriores ao ponto que está sendo analisado na trajetória. Para isso podem ser analisadas as regiões que compuseram a trajetória nos quadros $(T - l);\ l = 0, ..., k;\ k < T$, analogamente como o quadro T é analisado, por meio do modelo de aparência. Ou ainda analisar as regiões que compuseram a trajetória na sequência de quadros k_1 a k_2, onde $1 \leq k_1 \leq k_2 < T$;
- Exploração de outras formas de similaridade entre regiões correspondentes, ou seja, por meio da definição de um novo modelo de aparência, que pode ser dado utilizando, além da cor, características de textura;
- Desenvolver uma segmentação que permita ter *superpixels* consistentes de um quadro para outro.

Dentre as aplicações para o método proposto, podem ser citadas:

- Reconhecimento de padrões, realizando a detecção, monitoramento e análise do movimento. Pode ser realizado um agrupamento das trajetórias que representam um movimento conhecido e desta forma definir um padrão conhecido de movimento, que quando comparado a outro agrupamento de trajetórias de um outro vídeo, consiga distinguir que se trata do mesmo movimento ou não e dessa forma poder realizar a indexação e recuperação de vídeo baseada em conteúdo;

- Animações em realidade aumentada. Tendo conhecimento sobre como são as trajetórias do movimento é possível tornar os movimentos mais reais;
- Observação de atividades humanas. Podem ser feitas análises de eventos esportivos, coreografia de dança, entre outros;
- Monitoramento de tráfego. Analisando o fluxo do trânsito para detectar acidentes;
- Segurança. Detectando atividades suspeitas, por meio da diferença entre os padrões de trajetórias. A detecção do movimento e monitoramento do mesmo também pode ser usado utilizando informações de câmeras que capturam imagens com infravermelho;
- Terapia médica, para melhorar a qualidade de vida de pessoas que fazem terapia física e pessoas com deficiência, por meio do reconhecimento de gestos, definidos por trajetórias de movimentos conhecidos.

Referências Bibliográficas

[1] Ali, S., Basharat, A., and Shah, M. Chaotic invariants for human action recognition. *IEEE International Conference on Computer Vision (ICCV 2007)* (October 14-20 2007).

[2] Benary, W. *The Influence of form on Brightness Contrast*. W. D. Ellis, A Sourcebook of Gestalt Psychology, 1938.

[3] Besag, J. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological) 36*, 2 (1974), 192–236.

[4] Besag, J. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society B-48* (1986), 259–302.

[5] Blank, M., Gorelick, L., Shechtman, E., Irani, M., and Basri, R. Actions as space-time shapes. In *IEEE International Conference on Computer Vision (ICCV 2005)* (2005), pp. 1395–1402.

[6] Brox, T., Rosenhahn, B., Cremers, D., and peter Seidel, H. High accuracy optical flow serves 3-d pose tracking: exploiting contour and flow based constraints. In *European Conference on Computer Vision (ECCV 2006)* (2006), vol. 3952 of LNCS, Springer, pp. 98–111.

[7] Buzan, D., Sclaroff, S., and Kollios, G. Extraction and clustering of motion trajectories in video. In *IEEE International Conference on Pattern Recognition (ICPR 2004)* (August 2004), vol. 2, pp. 521–524.

[8] Compston, A. Action recognition in the premotor cortex. By Vittorio Gallese, Luciano Fadiga, Leonardo Fogassi and Giacomo Rizzolatti. Brain 1996: 119; 593-609. *Brain 132*, 7 (2009), 1685–1689.

[9] Dahme, G., Ribeiro, E., and Bush, M. Spatial statistics of textons. In *International Conference of Computer Vision Theory and Applications - VISAPP* (Setubal, Portugal, 2006).

[10] Filipovych, R., and Ribeiro, E. Learning human motion models from unsegmented videos. In *IEEE International Conference on Computer Vision and Patter Recognition (CVPR 2008)* (2008).

[11] Filipovych, R., and Ribeiro, E. Learning structural models in multiple projection spaces. In *IEEE International Conference on Image Analysis and Recognition (ICIAR 2009)* (Berlin, Heidelberg, 2009), Springer-Verlag, pp. 616–626.

[12] Fossati, A., Dimitrijevic, M., Lepetit, V., and Fua, P. Bridging the gap between detection and tracking for 3d monocular video-based motion capture. In *Conference on Computer Vision and Pattern Recognition* (Minneapolis, MI, June 2007).

[13] Gonzalez, R. C., and Woods, R. E. *Digital Image Processing (2nd Edition)*. Prentice Hall, January 2002.

[14] Gorelick, L., Blank, M., Shechtman, E., Irani, M., and Basri, R. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*, 12 (December 2007), 2247–2253.

[15] Ha, J., Alvino, C., Pryor, G., Niethammer, M., Johnson, E., and Tannenbaum, A. Active contours and optical flow for automatic tracking of flying vehicles. In *Proceedings of the American Control Conference* (2004), vol. 4, pp. 3441–3446.

[16] Jiang, H., Drew, M., and Li, Z.-N. Matching by linear programming and successive convexification. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*, 6 (June 2007), 959–975.

[17] Jiang, H., and Martin, D. R. Finding actions using shape flows. In *European Conference on Computer Vision (ECCV 2008)* (Berlin, Heidelberg, 2008), Springer-Verlag, pp. 278–292.

[18] Julesz, B. Textons, the elements of texture perception, and their interactions. *Nature 290*, 5802 (March 1981), 91–97.

[19] Ke, Y., Sukthankar, R., and Hebert, M. Spatio-temporal shape and flow correlation for action recognition. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2007)* (June 2007), pp. 1–8.

[20] Kim, I. Y., and Yang, H. S. An integrated approach for scene understanding based on markov random field model. *Pattern Recognition 28*, 12 (1995), 1887 – 1897.

[21] Kuhn, H. W. The Hungarian method for the assignment problem. *Naval Research Logistic Quarterly 2* (1955), 83–97.

[22] Lameira, A. P., Gawryszewski, L. d. G., and Pereira Jr., A. Neurônios espelho. *Psicologia USP 17* (12 2006), 123 – 133.

[23] Leung, T., and Malik, J. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision 43*, 1 (June 2001), 29–44.

[24] Levinshtein, A., Sminchisescu, C., and Dickinson, S. Multiscale symmetric part detection and grouping. In *IEEE International Conference on Computer Vision (ICCV 2009)* (September 2009), pp. 2162–2169.

[25] Levinshtein, A., Stere, A., Kutulakos, K. N., Fleet, D. J., Dickinson, S. J., and Siddiqi, K. Turbopixels: Fast superpixels using geometric flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence 31*, 12 (2009), 2290–2297.

[26] Lin, D., Grimson, W. E. L., and III, J. W. F. Learning visual flows: A lie algebraic approach. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2009)* (2009), IEEE, pp. 747–754.

[27] Little, J., and Boyd, J. E. Recognizing people by their gait: The shape of motion. *Videre 1* (1998), 1–32.

[28] Malik, J., Belongie, S., Leung, T. K., and Shi, J. Contour and texture analysis for image segmentation. *International Journal of Computer Vision 43*, 1 (2001), 7–27.

[29] Malik, J., Belongie, S., Shi, J., and Leung, T. Textons, contours and regions: Cue integration in image segmentation. In *IEEE International Conference on Computer Vision (ICCV 1999)* (1999), pp. 918–925.

[30] Matikainen, P., Hebert, M., and Sukthankar, R. Trajectons: Action recognition through the motion analysis of tracked features. In *Workshop on Video-Oriented Object and Event Classification* (September 2009), pp. 514–521.

[31] Meyn, S., and Tweedie, R. L. *Markov Chains and Stochastic Stability*. Cambridge University Press, New York, NY, USA, 2009.

[32] Mori, G. Guiding model search using segmentation. In *IEEE International Conference on Computer Vision (ICCV 2005)* (2005), vol. 2, pp. 1417–1423.

[33] Mori, G., Ren, X., Efros, A. A., and Malik, J. Recovering human body configurations: Combining segmentation and recognition. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2* (2004), 326–333.

[34] Natarajan, P., and Nevatia, R. View and scale invariant action recognition using multiview shape-flow models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008)* (June 2008), pp. 1–8.

[35] Nattkemper, T. W., Wersing, H., Ritter, H., and Schubert, W. Fluorescence micrograph segmentation by gestalt-based feature binding. *International Joint Conference on Neural Networks 1* (2000), 348–353.

[36] Niethammer, M., Tannenbaum, A., and Angenent, S. Dynamic active contours for visual tracking. *IEEE Transactions on Automatic Control 51*, 4 (April 2006), 562–579.

[37] Pardowitz, M., Haschke, R., Steil, J., and Ritter, H. Gestalt-based action segmentation for robot task learning. In *IEEE-RAS International Conference on Humanoid Robots, 2008* (Dec. 2008), pp. 347–352.

[38] Pelechano, N. Crowd simulation incorporating agent psychological models, roles and communication. In *First International Workshop on Crowd Simulation* (2005), pp. 21–30.

[39] Porikli, F. Trajectory pattern detection by hmm parameter space features and eigenvector clustering. In *European Conference on Computer Vision (ECCV 2004)* (May 2004), pp. 1–8.

[40] Rasmussen, C., Lu, Y., and Kocamaz, M. Appearance contrast for fast, robust trail-following. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2009), pp. 3505–3512.

[41] Reed, T., and Wechsler, H. Segmentation of textured images and gestalt organization using spatial/spatial-frequency representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence 12*, 1 (1990), 1–12.

[42] Ren, X., and Malik, J. Learning a classification model for segmentation. In *IEEE International Conference on Computer Vision (ICCV 2003)* (2003), vol. 1, pp. 10–17.

[43] Rizzolatti, G., Fadiga, L., Gallese, V., and Fogassi, L. Premotor cortex and the recognition of motor actions. *Cognitive Brain Research 3*, 2 (1996), 131 – 141. Mental representations of motor acts.

[44] Sato, K., and Aggarwal, J. K. Temporal spatio-velocity transform and its application to tracking and interaction. *Computer Vision and Image Understanding 96*, 2 (2004), 100–128.

[45] Serby, D., Meier, E., and Van Gool, L. Probabilistic object tracking using multiple features. In *IEEE International Conference on Pattern Recognition (ICPR 2004)* (Aug. 2004), vol. 2, pp. 184–187.

[46] Shi, J., and Malik, J. Normalized cuts and image segmentation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 1997)* (Washington, DC, USA, 1997), IEEE Computer Society, p. 731.

[47] Shi, J., and Malik, J. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22* (2000), 888–905.

[48] Trémeau, A., and Colantoni, P. Regions adjacency graph applied to color image segmentation. *IEEE Transactions on Image Processing 9* (2000), 735–744.

[49] Veenman, C., Reinders, M., and Backer, E. Resolving motion correspondence for densely moving points. *IEEE Transactions on Pattern Analysis and Machine Intelligence 23*, 1 (Jan 2001), 54–72.

[50] Vela, P. A., Niethammer, M., Malcolm, J., and Tannenbaum, A. Closed loop visual tracking using observer-based dynamic active contours. In *Proceedings of the Conference on Guidance Navigation and Control* (2005).

[51] Wertheimer, M. *Laws of organization in perceptual forms.* W. D. Ellis, A Source-book of Gestalt Psychology, 1938.

[52] Xin, A. Y., Li, X., and Shah, M. Object contour tracking using level sets. In *Asian Conference on Computer Vision, ACCV 2004, Jaju Islands, Korea* (2004).

[53] Yilmaz, A., Javed, O., and Shah, M. Object tracking: A survey. *ACM Computing Surveys (CSUR) 38*, 4 (2006).

[54] Zhang, T., Lu, H., and Li, S. Learning semantic scene models by object classification and trajectory clustering. In *IEEE International Conference on Computer Vision and Patter Recognition (CVPR 2009)* (2009), pp. 1940–1947.

[55] Zhao, T., Nevatia, R., and Wu, B. Segmentation and tracking of multiple humans in crowded environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence 30*, 7 (2008), 1198–1211.