Roberta Geneci Neves Weber Teigão

Análise de Vídeo por Ritmo Visual e Morfologia em Cores

Curitiba

Novembro de 2007

Roberta Geneci Neves Weber Teigão

Análise de Vídeo por Ritmo Visual e Morfologia em Cores

Dissertação de Mestrado submetida ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para a obtenção do título de Mestre em Informática.

Área de concentração: *Ciência da Computação*

Orientador: Prof. Dr. Jacques Facon

Curitiba

Novembro de 2007

Esta folha deve ser substituída pela ata de defesa devidamente assinada, que será fornecida pela secretaria do programa após a defesa.

Teigão, Roberta Geneci Neves Weber
Análise de Vídeo por Ritmo Visual e Morfologia em Cores. Curitiba, 2007. 87pp.
Dissertação (Mestrado) - Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática.
1. Ritmo Visual 2. Morfologia em Cores 3. Segmentação de Vídeo 4. Vídeo Comprimido. I. Pontifícia Universidade Católica do Paraná. Centro de Ciências Exatas e de Tecnologia. Programa de Pós-Graduação em Informática II-t.

Com amor, ao meu marido Rafael, por ter me despertado a vontade de iniciar os estudos de Mestrado.

Agradecimentos

Agradeço, primeiramente, a Deus por me acompanhar em todos os momentos desta conquista.

Agradeço ao Professor Dr. Jacques Facon, por toda amizade e dedicação, mas, principalmente, por ter aceitado o desafio de prosseguir com a minha orientação no momento em que fiquei sem um orientador.

Agradeço aos meus pais, Alvaro e Cida, por todo o apoio e por suas valiosas correções ortográficas finais.

Agradeço ao meu marido, Rafael, por todo carinho, paciência, ajuda e palavras de motivação nos momentos de desânimo.

Por fim, agradeço a minha família pelo incentivo, e em especial a Arabela por sempre nos lembrar da importância de se tornar Mestre.

“A mente que se abre a uma nova idéia
jamais voltará ao seu tamanho original.”

Albert Einstein

Sumário

Lista de Figuras

Lista de Tabelas

Resumo

Com o avanço da tecnologia digital e o conseqüente crescimento na utilização de vídeos digitais, aumenta-se a necessidade de recuperação de informação de interesse nesta mídia que apresenta enormes volumes de dados. Muitas pesquisas sobre indexação e processamento de vídeo digital têm sido realizadas em busca de consultas eficientes e recuperação de conteúdo. Neste contexto, o problema da detecção de transições entre tomadas é o primeiro passo para a segmentação e análise de vídeo digital, e será o objeto de estudo da presente pesquisa, que trará algumas contribuições para a abordagem chamada ritmo visual por amostragem. Neste trabalho, a análise de vídeo será realizada sobre uma imagem formada a partir da diagonal principal de uma versão reduzida de cada quadro, utilizando-se a morfologia em cores para detecção das transições, sem aplicar a descompressão prévia do vídeo e sua conversão para níveis de cinza. Desta forma, a metodologia proposta não desconsidera a importante informação presente na cor e, além disso, realiza o processamento de uma quantidade de informação muito menor. O método proposto foi testado em 15 vídeos de diferentes tipos, obtendo-se, em média, valores de 78% e 81%, de precisão e revocação, respectivamente.

Palavras-chave: ritmo visual, segmentação de vídeo, morfologia em cores, vídeo comprimido.

Abstract

With the advance of digital technology and consequent increase in the use of the digital videos, the necessity for recovering interesting information in this enormous volume of data increases. Many researches on digital video indexing and processing have been made in order to efficiently query and recover content. In this context, the problem of detecting transitions among shots is the first step for digital video segmentation and analysis. This will be the object of study of the present research, which will bring some contributions on the approach called visual rhythm. In this work, the video analysis will be made through an image formed by the main diagonal of a reduced version of each frame, using colored morphology for transitions' detection without applying the previous decompression and conversion for grayscale levels. Thus, the proposed methodology does not reject the important information present in the colors; further more, it processes lesser amount of information. The proposed methodology was tested in 15 different videos, getting, in average, values between 78% and 81% of precision and recall respectively.

Keywords: visual rhythm, video segmentation, colored morphology, compressed video.

Capítulo 1

Introdução

Nas últimas décadas, a televisão analógica teve uma influência fundamental na disseminação da informação. Porém, esta tecnologia possui algumas limitações tais como: dificuldades de edição e de controle sobre a qualidade do vídeo, restrições na criação de aplicações com interatividade e dificuldades de localização de imagens no vídeo.

Com o desenvolvimento da informática e eletrônica, ocorreu uma grande transformação nos meios de comunicação, que passaram a migrar dos formatos analógicos para os formatos digitais.

Na digitalização de um vídeo, cada quadro é transformado em *pixels*, ou seja, a informação da cor de cada ponto da imagem é armazenada em um *pixel*. A qualidade de cada quadro depende da quantidade de *pixels* utilizados e da quantidade de informações em cada *pixel* [37].

Diferentes métodos para a compactação e transmissão de vídeo digital têm sido desenvolvidos, comprimindo-se dados redundantes, reduzindo-se espaço de armazenamento e de banda para a transmissão.

De maneira geral, o vídeo digital oferece maior qualidade de gravação, facilidade de processamento, eficiência em termos de banda e a possibilidade de criação de aplicações com interatividade. Estas vantagens aliadas à facilidade de compartilhamento e de gravação através das câmeras digitais, têm tornado o vídeo digital cada vez mais popular.

Com todo o avanço da tecnologia digital, a disponibilidade de conteúdo multimídia cresce a cada dia. Aplicações como bibliotecas digitais, ensino à distância, vídeo sob demanda, transmissão de vídeo digital e sistemas de informação multimídia são exemplos da vasta coleção disponível.

Junto à disponibilidade, cresce, também, a necessidade de consultas e busca de informação relevante neste enorme volume de dados. Muitas pesquisas surgem para tentar aprimorar as técnicas de indexação, pesquisa, busca e recuperação (*retrievel*) de informação em bancos de dados de vídeos.

A recuperação de conteúdo de interesse em vídeos está, geralmente, associada à indexação manual de informações, tornando-se um processo inadequado para grandes volumes de vídeo. Além disso, para uma indexação eficiente, deve haver a identificação e a compreensão das unidades fundamentais do vídeo. Desta forma, a indexação torna-se uma tarefa difícil, haja vista o comprimento, generalidade de conteúdo e formato não estruturado dos vídeos [16]. Revela-se, então, a necessidade de processos automáticos para a indexação de vídeo, que possibilitem buscas rápidas e eficientes.

Muitas pesquisas sobre a análise de conteúdo de vídeo têm sido realizadas e diferentes abordagens têm sido propostas. Geralmente, as ferramentas de análise de vídeo possuem as seguintes etapas de processamento [16]:

***video parsing*:** o processo de *video parsing* consiste na segmentação do vídeo em unidades fundamentais tais como tomadas e cenas, no reconhecimento de operações de câmera e na identificação dos quadros-chaves.

sumarização: a técnica da sumarização é usada para resumir o conteúdo de uma seqüência de vídeo, facilitando o seu acesso (navegação e recuperação).

classificação: a classificação tem como objetivo classificar o vídeo, a cena ou a tomada em diferentes categorias.

indexação: o processo indexação associa termos descritivos ao vídeo, visando facilitar consultas em grandes bancos de dados.

Antes da sumarização, classificação e indexação de vídeos digitais é necessário, primeiramente, detectar-se as mudanças de tomadas presentes na seqüência de vídeo [23]. Sendo assim, o *video parsing* é o primeiro passo na análise e segmentação de vídeo e será o objeto de estudo da presente pesquisa.

A maioria das técnicas de *video parsing* existentes na literatura empregam medidas de dissimilaridade entre quadros sucessivos, baseadas em informações de cor, forma e textura para a detecção de transições. Diferentemente deste tipo de abordagem, nesta dissertação a análise

do vídeo será realizada sobre uma imagem 2D formada a partir de fatias de cada quadro do vídeo, como será melhor descrita na seção de 3.4.

1.1 Desafios

O desafio da presente dissertação é desenvolver uma nova metodologia para a detecção automática de transições abruptas de vídeo (cortes), empregando-se as técnicas de ritmo visual por amostragem (seção 3.4) e morfologia em cores (seção 2.6), trabalhando-se no domínio de compressão. Fazendo, assim, o uso das vantagens proporcionadas em se trabalhar diretamente com o vídeo comprimido (MPEG - *Moving Picture Expert Group*), tais como: a criação do ritmo visual a partir de miniaturas de quadros e a utilização de algumas informações disponíveis no MPEG para auxiliar na detecção.

O desafio deste trabalho é, sobretudo, realizar um estudo sobre segmentação de vídeo voltada a uma base de testes realista, composta de jogos de futebol, com todas as dificuldades que este tipo de vídeo oferece, enumerando-se os problemas encontrados e as soluções propostas.

1.2 Motivação

A segmentação de vídeo é uma etapa primordial no processamento do vídeo digital. Sendo assim, é indispensável o desenvolvimento e aperfeiçoamento das técnicas existentes visando rapidez de processamento e eficiência nas detecções para o uso em aplicações em tempo real.

As abordagens anteriores, que utilizavam a análise de imagens 2D para a detecção de transições, possuíam um alto custo computacional e resultados não satisfatórios [16]. Já a abordagem de Guimarães [16] não é adaptada às necessidades atuais, pois exige a prévia descompressão do vídeo e sua inteira conversão para níveis de cinza.

A presente pesquisa visa utilizar as vantagens em se trabalhar no domínio de compressão, tais como o uso de miniaturas de quadros, que possibilitam o processamento de uma quantidade muito menor de informação, objetivando rapidez de processamento. Além disso, objetiva a eficiência nas detecções com o uso das informações de cores, descartadas em abordagens anteriores.

1.3 Proposta

O presente trabalho tem como objetivo realizar a detecção de transições abruptas de vídeo, partindo da abordagem de Ngo [29], que utiliza fatias espaço-temporais para obtenção de uma simplificação do vídeo, transformando-o em uma imagem 2D. A idéia desta técnica é extrair uma fatia diagonal (ou coluna central, ou linha horizontal) de cada quadro para compor uma linha vertical em uma imagem. Sobre esta imagem, formada pelas amostras de cada quadro, é realizada a análise de vídeo, identificando-se os padrões de transições. Para a identificação destas transições, Ngo adota uma metodologia complexa que propõe modelos de energia baseados na descontinuidade de cor e textura, exigindo um alto custo computacional de processamento.

O trabalho proposto por Guimarães [16] também emprega a análise de vídeo a partir de uma imagem 2D, denominada de *ritmo visual por amostragem*, e utiliza a morfologia matemática em níveis de cinza para a identificação dos padrões de cada transição. Porém, esta técnica exige a descompressão prévia do vídeo e a sua conversão para níveis de cinza, o que limita o seu uso em sistemas em que o tempo é escasso.

O objetivo principal desta pesquisa é simplificar as abordagens apresentadas por Ngo e Guimarães para identificação de cortes, empregando-se a morfologia em cores para a detecção dos padrões. A imagem de ritmo visual é criada e trabalhada em cores. A conversão para níveis de cinza só ocorre depois da detecção de bordas, quando a quantidade de informações é muito menor. A utilização da morfologia em cores para a identificação dos padrões de cortes visa reduzir as perdas ocorridas ao se trabalhar com níveis de cinza. Para empregar a morfologia em cores, a ordenação proposta por Calixto [9] é utilizada.

Ao contrário da abordagem de Guimarães, este trabalho utiliza vídeos comprimidos em formato MPEG, trabalhando-se diretamente sobre o domínio de compressão e imagens em cores. De forma que o ritmo visual colorido é criado rapidamente, possibilitando o uso desta abordagem em sistemas em tempo real.

Tendo em vista a escolha pelo formato MPEG, a técnica proposta utiliza algumas das facilidades da disponibilização de informações no *stream* do vídeo. As detecções são realizadas sobre imagens de ritmo visual formadas a partir de miniaturas de quadros (imagens DC, seção 2.4), possibilitando uma maior rapidez de processamento, visto que a quantidade de informação processada é muito menor.

1.4 Contribuição

A principal contribuição desta pesquisa é comprovar que a nova metodologia proposta para a detecção de cortes permite rapidez de processamento e eficiência nas detecções. Da mesma forma, demonstra que a utilização da cor é um elemento importante na detecção de padrões e a aplicação da morfologia em cores, baseada na constante de cromaticidade, é eficiente nas detecções de cortes.

Além disso, revela que a utilização do vídeo comprimido pode ser vantajosa em relação à rapidez de processamento e facilidade de detecções, pois o MPEG possibilita a utilização de miniaturas de quadros e o emprego das informações relativas aos quadros I (*intra frame*, seção 2.4) no auxílio nas detecções.

Esta pesquisa visa, ainda, contribuir com um estudo sobre temas não abordados em outros trabalhos, mas não menos importantes para a eficiência da metodologia, tais como: a influência da largura da amostra na eficiência da detecção, as principais dificuldades na busca pelos padrões de corte e a melhor opção de amostra de quadro, dentre a vertical, horizontal ou diagonal, de acordo com a base de jogos de futebol.

1.5 Organização

Esta dissertação é organizada como nos 7 capítulos a seguir:

Capítulo 1 – Introdução O capítulo 1 possui uma breve introdução sobre a importância da segmentação de vídeo, desafios, motivação, proposta e contribuição desta dissertação.

Capítulo 2 – Fundamentação Teórica O capítulo 2 apresenta uma fundamentação teórica indispensável para compreender os assuntos tratados nesta dissertação, tais como vídeo digital, transições de vídeo, operações de câmera, vídeo comprimido, morfologia matemática, morfologia em cores e limiarização.

Capítulo 3 – Estado da Arte O capítulo 3 fornece o estado da arte atual, descrevendo as metodologias de segmentação de vídeo existentes.

Capítulo 4 – Metodologia O capítulo 4 descreve, detalhadamente, toda a metodologia utilizada na abordagem proposta, desde a montagem do ritmo visual até os processamentos empregados para a detecção dos cortes.

Capítulo 5 – Experimentos e Resultados No capítulo 6 são fornecidos os procedimentos necessários na realização dos testes. Além disso, fundamenta desde a escolha da base até a metodologia empregada na avaliação dos resultados. Fornece, ainda, os resultados obtidos com a implementação da metodologia proposta sobre a base de jogos de futebol e sobre outra base.

Capítulo 6 – Análise de Resultados No capítulo 6 são descritos detalhes na análise de resultados, bem como os principais desafios encontrados na detecção dos cortes.

Capítulo 7 – Conclusões e Trabalhos Futuros Este capítulo apresenta as principais conclusões sobre a abordagem proposta e indica os futuros trabalhos a respeito desta metodologia que podem complementar a pesquisa.

Capítulo 2

Fundamentação Teórica

Neste capítulo são apresentados alguns conceitos sobre vídeo digital que serão utilizados no restante do trabalho. Um vídeo digital é formado por quadros, tomadas e cenas, conforme pode ser observado na figura 2.1.

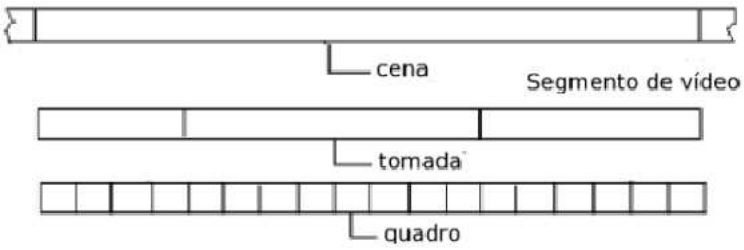

Figura 2.1: Estrutura de um segmento de vídeo digital

2.1 Vídeo digital

Definição 2.1. Vídeo "Um vídeo é uma mídia para armazenamento e transmissão de informações" [16].

Definição 2.2. Quadro Um quadro é o menor elemento da estrutura de um vídeo digital. Quadro-chaves, no MPEG, são um ou mais quadros que possuem a melhor representação do conteúdo de uma tomada ou uma cena.

Definição 2.3. Tomada "Uma tomada consiste de um ou mais quadros, gerados e gravados ininterruptamente, representando uma ação contínua em relação ao tempo e espaço" [12].

Definição 2.4. Cena "Uma cena é definida em termos de sua semântica, ou seja, um grupo de tomadas que se caracterizam pela mesma idéia" [40]. Um vídeo completo é composto por uma ou mais cenas.

2.2 Transições

Uma seqüência de vídeo pode possuir dois tipos de transição entre suas tomadas. A transição pode ser realizada, simplesmente, pela concatenação dos quadros entre as tomadas ou pela inserção de efeitos de edição, formando uma passagem gradual de uma tomada para outra.

Transições graduais são menos freqüentes e mais difíceis de se detectar que transições abruptas. Estas devem ser diferenciadas das operações da câmera e movimentos de objeto, pois estes causam falsas detecções.

Definição 2.5. Transições abruptas As transições abruptas são os cortes, ou seja, duas tomadas são concatenadas sem modificação ou criação de nenhum quadro entre elas. É o tipo mais simples de transição. A figura 2.2 ilustra um corte em um segmento de vídeo.

Figura 2.2: Exemplo de corte

Definição 2.6. Transições graduais "As transições graduais são efeitos de edição que podem ser aplicados para combinar duas tomadas de câmera, criando gradualmente uma transição" [40]. Alguns quadros são artificialmente criados ou modificados na transição gradual.

Definição 2.7. *Fade-out* "Um fade-out é um decréscimo gradual da luminosidade dos quadros de uma tomada até resultar em um quadro preto" [22]. Afigura 2.3 na página seguinte é um exemplo de *fade-out*.

Definição 2.8. *Fade-in* Um fade-in é um acréscimo gradual da luminosidade dos quadros de uma tomada, começando por um quadro preto até a obtenção de um quadro com luminosidade natural. A figura 2.4 na próxima página ilustra um f*ade-in.*

Figura 2.3: Exemplo de *fade-out*

Figura 2.4: Exemplo de *fade-in*

Definição 2.9. Dissolve Um dissolve é uma transição na qual dois quadros pertencentes a tomadas diferentes são misturados. A medida que os quadros da primeira tomada começam a perder seus pixels e a desaparecer, os quadros da segunda tomada começam a ganhar pixels até obter seu conteúdo completo, substituindo o original. No dissolve é como se um *fade-out* ocorresse na primeira tomada simultaneamente a um *fade-in* da segunda tomada [22]. A figura 2.5 ilustra um dissolve.

Figura 2.5: Exemplo de *dissolve*

Definição 2.10. *Wipe* Um wipe é uma transição na qual uma linha horizontal (ou vertical) delimita duas tomadas e movimenta-se gradualmente em uma direção até o aparecimento total da nova tomada. A figura 2.6 na próxima página ilustra um *wipe* horizontal.

Figura 2.6: Exemplo de *wipe* horizontal

Outro evento muito comum em seqüências vídeos é o *flash*. *Flashes* são efeitos caracterizados pelo aumento da luminosidade em alguns quadros da seqüência de vídeo e é muito utilizado em jornais televisivos.

Figura 2.7: Exemplo de *flash*

2.3 Operações de câmera

Durante as filmagens de um vídeo, efeitos chamados de operações de câmera podem ser, ainda, utilizados. Estes recursos podem ser obtidos pela movimentação da câmera, mudança de ângulo de filmagem ou realização de *zoom* e *pan* como na figura 2.8.

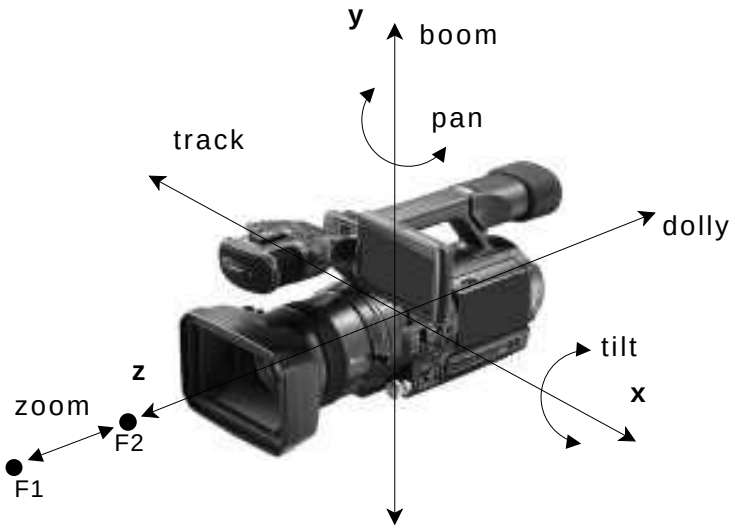

Figura 2.8: Operações de câmera: *zoom*, *pan*, *tilt*, *track*, *boom* e *dolly*

A operação de *zoom* corresponde a uma mudança da distância focal (de F1 para F2). O *pan* é definido como uma rotação de câmera em torno do eixo y e o *tilt* corresponde à rotação

da câmera em torno do eixo x. *Track* é o movimento transversal horizontal no eixo x e *boom* é o movimento transversal vertical no eixo y. A operação *dolly* corresponde ao movimento lateral horizontal no eixo z.

O reconhecimento de operação de câmera é um fator importante, tendo em vista a possibilidade de perceber para onde a atenção do espectador está sendo direcionada, indicando, deste modo, a seleção dos quadros-chaves. Por exemplo, quando um panorama é filmado, a seqüência inteira pertence à mesma tomada, porém como o conteúdo da seqüência muda substancialmente, o conteúdo deve ser substituído por mais de um quadro-chave. Quando operações de *zooms* são realizadas, toda a tomada pode ser representada por dois quadros: o inicial e o final [22].

2.4 Vídeo comprimido

Devido as limitações de tempo e espaço, a maioria das imagens e vídeos disponíveis encontra-se em formato comprimido. Desta forma, é extremamente importante que as metodologias de análise de vídeo trabalhem diretamente sobre vídeo comprimido.

O padrão *Moving Picture Expert Group* (conhecido como MPEG) é o padrão mais aceito internacionalmente para a compressão de vídeo digital. Muito usado nas TVs digitais, leitores de DVD, vídeo conferência, decodificadores HDTV (*High Definition Television*), entre outras aplicações. A família MPEG, estabelecida pela União Internacional de Telecomunicações (ITU), possui algumas vantagens sobre os demais padrões, tais como a compatibilidade universal, altas taxas de compressão e perda aceitável na qualidade final [41].

O MPEG-1 é o padrão MPEG inicial, finalizado em 1991 e otimizado para se trabalhar com a taxa de 1.5 Mbps e resolução de 352x240 *pixels* [33].

O MPEG-2, finalizado em 1994, possui taxas de compressão de 3Mbps a 100Mbps, suportando arquivos dados maiores. Este é o padrão utlizado em aparelhos de DVD e muitos sistemas de televisão digital. O MPEG-3 foi criado para a sua utilização em HDTV (*High Definition Television*), porém este padrão foi abandonado, pois o MPEG-2 supria o volume de dados exigido pelo HDTV [5].

A criação do MPEG-4 visa adaptar-se melhor à Internet, permitindo uma transmissão com qualidade superior ao MPEG-1 com uma taxa de bits menor. Por fim, o padrão MPEG-7 não é um formato de codificação de vídeo, mas um padrão para a descrição de objetos multimídia [5].

O MPEG faz uso de dois tipos de redundância: a espacial e a temporal. A espacial se refere a redundância em uma mesma imagem e a temporal diz respeito à redundância em quadros consecutivos. Neste contexto, o padrão MPEG trabalha aplicando dois tipos de compressão: a compensação de movimento para reduzir redundância temporal e a compressão baseada em bloco para reduzir a redundância espacial.

O MPEG define três tipos de quadros: I (*intra frame*), P (*forward predicted frame*s) e B (*bidirectionally predicted frame*s) que são combinados em um padrão repetitivo chamado GOP *(group of pictures*). A figura 2.9 mostra o modelo MPEG de compensação de movimento.

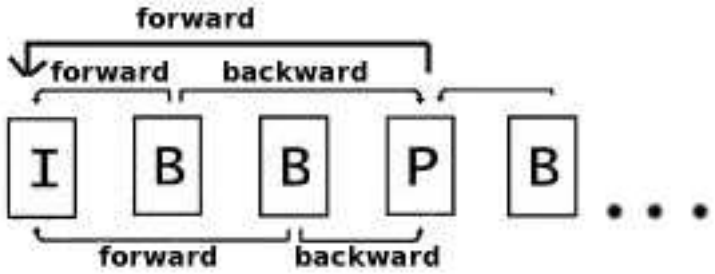

Figura 2.9: Exemplo do padrão GOP

Os quadros I são quadros que são codificados individualmente sem nenhuma predição temporal, usando apenas informação presente na imagem pela Transformada Discreta do Cosseno (DCT), Quantização, *Run Length Encodin*g (RLE) e codificação de Huffman. Os quadros P são os quadros preditos à frente, codificados com compensação de movimento usando o quadro precedente mais próximo (quadro I ou P). Os quadros B também usam a compensação de movimento e possuem codificação relativa ao quadro de referência precedente ou sucessivo, ou ambos [22].

Para eliminar a redundância espacial, a compressão é baseada nas cores, utilizando a DCT. O olho humano é muito sensível a variação de cor, mas a sua interpretação no cérebro é mais caracterizada pela luminosidade. Assim, para tirar vantagem desta propriedade da visão humana, o MPEG utiliza o espaço de cores YUV (Y - luminância, U e V - componentes de cores). O algoritmo busca agrupamentos de *pixels* com a mesma cor e os substitui por um único código [33].

Os quadros I são divididos em blocos de 8 x 8 *pixels* e a cada bloco é aplicada a DCT, que transforma dados de amplitude para freqüência. Os coeficientes da DCT são quantizados para reduzir sua amplitude e aumentar o número de coeficientes iguais a zero, podendo, assim, descartar a informação que é visualmente insignificante [33]. O primeiro coeficiente DCT é

chamado termo DC e representa a média do respectivo bloco.

Muitas abordagens de segmentação de vídeo digital utilizam imagens construídas a partir dos termos DC, chamadas imagens DC. Estas representam uma miniatura de cada quadro, pois são construídas a partir dos termos DC que representam a média de cada bloco.

Os coeficientes DCT quantizados são, então, realocados no padrão zig-zag em que as baixas freqüências são seguidas pelas altas, visando aumentar o número de coeficientes consecutivos iguais a zero em cada bloco. Os coeficientes DC quantizados são codificados pelo número de bits significativos, seguido pelos próprios bits. Finalmente, o código de Huffman (compressão baseada na Entropia dos dados) é aplicado [33].

Para reduzir a redundância temporal, o MPEG utiliza a compensação de movimento baseada em bloco aplicada para quadros P e B. A imagem é dividida em macroblocos (MB) de 16 x 16 *pixels* e apenas um vetor de movimento (MV) é estimado, codificado e transmitido para cada um destes blocos, ou seja, ao invés de enviar toda a imagem a cada quadro, o MPEG transmite apenas as diferenças do novo quadro em relação ao anterior.

2.5 Morfologia Matemática

A Morfologia Matemática estuda a forma e a estrutura geométrica dos objetos de uma imagem, visando revelar informações referentes a sua geometria e topologia, através da execução de certas operações matemáticas sobre seus *pixels* [39]. Estas operações são realizadas por meio de operadores morfológicos e elementos estruturantes.

Um elemento estruturante constitui-se de um certo conjunto de *pixels* (de tamanho e forma conhecidos) utilizado para se comparar com os *pixels* de uma imagem durante a execução de um operador morfológico. Se o elemento estruturante coincidir com alguma estrutura da imagem, então uma transformação é aplicada. Desta forma, o formato e tamanho do elemento estruturante influencia nas propriedades geométricas que serão extraídas da imagem [2].

2.5.1 Introdução

A Morfologia Matemática visa auxiliar na solução de vários problemas de processamento de imagens e visão computacional como filtragem, detecção de bordas, segmentação, realce, afinamento, entre outros.

A Morfologia Matemática divide-se em binária (aplicada para imagens binárias), em

níveis de cinza e em cores (aplicada para imagens em tons de cinza e coloridas). As operações morfológicas binárias buscam uma determinada estrutura de *pixels* pretos e brancos sobre a vizinhança ao redor de cada *pixel* da imagem, e este deve ser ativado ou desativado de acordo com a operação. As operações morfológicas em níveis de cinza e cores buscam o valor de *pixel* mais escuro ou mais claro dentro da vizinhança de um *pixel* central, substituindo-o pelo valor máximo ou mínimo, de acordo com a operação [14].

Segundo Facon, as operações fundamentais da Morfologia Matemática são a operações duais dilatação e erosão. Porém, muitas outras operações poderosas podem ser construídas pela interação destas duas operações básicas.

2.5.2 Erosão binária

A erosão binária **ero** de um conjunto X pelo elemento estruturante *B* é definida pela seguinte equação 2.1.

$$ero^B(X) = X\ ero\ B = \{x \in \epsilon : B_x \subset X\} \tag{2.1}$$

A operação de erosão binária ocorre deslizando-se o elemento estruturante *B* sobre toda a imagem *X*. O *pixel x* corresponde ao ponto central do elemento estruturante e este é ativado, se o elemento estruturante coincidir totalmente com a vizinhança de *x* na imagem original. De outra forma, ele é marcado como um *pixel* irrelevante na imagem do resultado [14].

A erosão modifica a imagem original, sempre diminuindo o conjunto inicial, pois os grupos de *pixels* inferiores ao elemento estruturante são eliminados, como pode ser constatado no exemplo da figura 2.10 na próxima página. Este exemplo mostra o resultado de uma erosão binária de uma imagem *X* pelo elemento estruturante *B,* cujo ponto central é indicado pelo asterisco.

2.5.3 Dilatação binária

A dilatação binária **dil** de um conjunto *X* pelo elemento estruturante *B* é definida pela seguinte equação 2.2.

$$dil^B(X) = X\ dil\ B = \{x \in X : B_x \cap X \neq \phi\} \tag{2.2}$$

Como na erosão, na operação de dilatação binária deve-se, também, deslizar o elemento

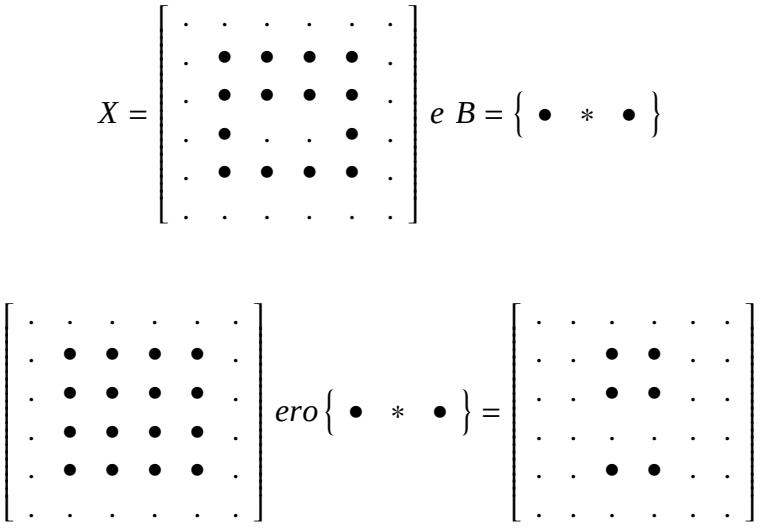

Figura 2.10: Erosão binária de X pelo elemento estruturante B.

estruturante *B* sobre toda a imagem *X*. O *pixel x* corresponde ao ponto central do elemento estruturante e este é ativado, se houver uma intersecção do elemento estruturante com a vizinhança de *x* na imagem original. De outra forma, ele é marcado como um *pixel* irrelevante na imagem de resultado [14].

A dilatação modifica a imagem original, sempre aumentando o conjunto inicial e preenchendo os furos menores que o elemento estruturante, como pode ser verificado no exemplo da figura 2.11. Este exemplo mostra o resultado de uma dilatação binária de uma imagem *X* pelo elemento estruturante *B*, cujo ponto central é indicado pelo asterisco.

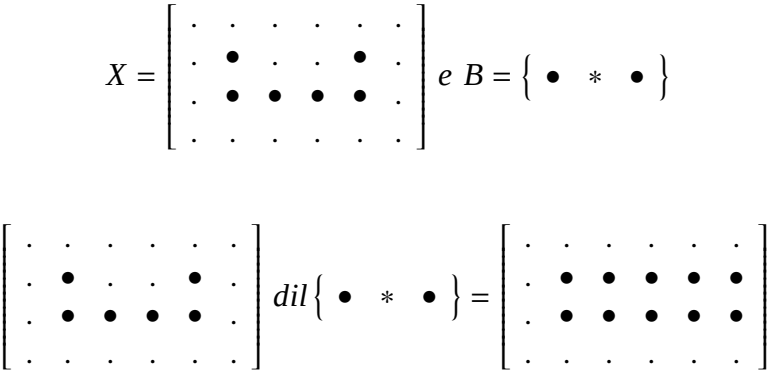

Figura 2.11: Dilatação binária de X pelo elemento estruturante B.

2.5.4 Condicionalidade

Existem situações em que há a necessidade de se diferenciar o processamento aplicado a uma imagem em função da geometria dos objetos. Os operadores de erosão ou dilatação condicionais tornam possível o processamento diferenciado de uma imagem, ou seja, permitem a definição de um subconjunto da imagem para o qual as operações são válidas.

Erosão condicional

A equação 2.3 define a erosão condicional de um subconjunto Z de X, também chamado de marcador, pelo elemento estruturante B em relação ao conjunto X, chamado de máscara [14].

$$ero_{cX}^{B}(Z) = (ero^{B}(Z \cup X^{c})) \cap X \tag{2.3}$$

Dilatação condicional

A equação 2.4 define a dilatação condicional de um subconjunto Z, também chamado de marcador, pelo elemento estruturante B, chamado de máscara [14].

$$dil_{cX}^{B}(Z) = dil^{B}(Z) \cap X \tag{2.4}$$

A dilatação condicional baseia-se em uma dilatação do subconjunto Z pelo elemento estruturante B, seguida de uma intersecção com o conjunto X, como pode ser verificado no exemplo da figura 2.12 na página seguinte. O ponto central do elemento estruturante B é representado pelo asterisco.

2.5.5 Erosão em níveis de cinza

A equação 2.5 representa a erosão de um sinal f por um elemento estruturante tridimensional g:

$$\varepsilon^{g}(f(x)) = Min\{f(y) - g(x - y) : y \in D[g]\} \tag{2.5}$$

onde Min equivale ao mínimo, x é o ponto a ser processado na imagem original, y são os pontos envolvidos pelo elemento estruturante e $D[g]$ é o domínio do elemento estruturante.

A erosão em níveis de cinza modifica a imagem original e de maneira geral, os seus

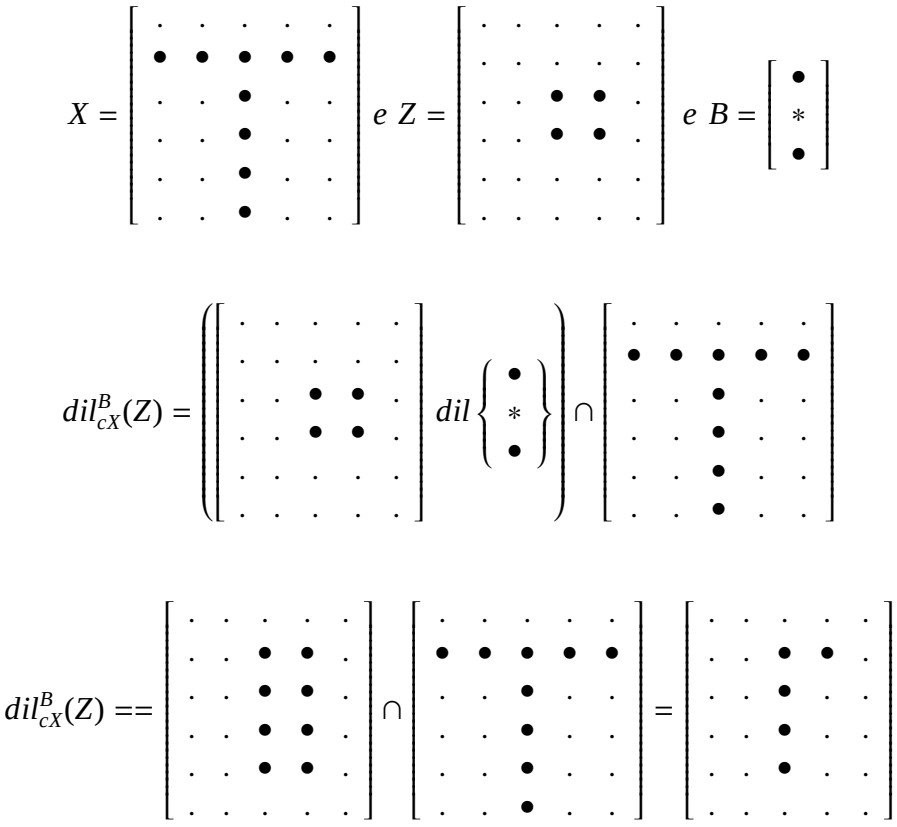

Figura 2.12: Dilatação condicional do subconjunto Z, segundo X, pelo elemento estruturante B.

efeitos são: escurecer a imagem, expandir os padrões escuros, reduzir e, até mesmo, eliminar padrões mais claros [14], conforme exemplificado na figura 2.13.

Figura 2.13: Erosão em níveis de cinza com elemento estruturante quadrado planar e duas iterações: (a) Imagem original, (b) Imagem erodida.

2.5.6 Dilatação em níveis de cinza

A equação 2.6 representa a dilatação de um sinal f por um elemento estruturante g:

$$\delta^{g}(f(x)) = Max\{f(y) + g(x-y) : y \in D[g]\} \tag{2.6}$$

onde Max equivale ao máximo, x é o ponto a ser processado na imagem original, y são os pontos envolvidos pelo elemento estruturante e $D[g]$ é o domínio do elemento estruturante.

De forma geral, os efeitos da dilatação em níveis de cinza são: clarear a imagem, expandir os padrões mais claros, reduzir e, até mesmo, eliminar os padrões escuros [14], conforme exemplificado na figura 2.14 na página seguinte.

2.5.7 Abertura em níveis de cinza

A operação de abertura em níveis de cinza baseia-se em uma erosão de um conjunto f por um elemento estruturante g, seguida de uma dilatação do conjunto erodido pelo mesmo elemento estruturante g, como mostrado na equação 2.7.

$$abe^{g}(f) = dil^{g}(ero^{\tilde{g}}(f)) \tag{2.7}$$

A equação 2.8 define a operação de abertura em níveis de cinza de um conjunto f pelo elemento estruturante g.

Figura 2.14: Dilatação em níveis de cinza com elemento estruturante quadrado planar e duas iterações: (a) Imagem original, (b) Imagem dilatada.

$$abe^g(f) = f\ abe\ g = (f \ominus \tilde{g}) \oplus g \tag{2.8}$$

De maneira geral, uma imagem aberta é mais regular e menos rica em detalhes que a imagem original. Os principais efeitos da abertura sobre uma imagem são: separar os padrões claros próximos, eliminar os padrões claros inferiores ao elemento estruturante, manter os padrões escuros afastados e conectar os padrões escuros mais próximos [14].

2.5.8 Fechamento em níveis de cinza

A operação de fechamento em níveis de cinza baseia-se em uma dilatação de um conjunto f por um elemento estruturante g, seguida de uma erosão do conjunto dilatado pelo mesmo elemento estruturante g, como mostrado na equação 2.9.

$$fec^g(f) = ero^g(dil^{\tilde{g}}(f)) \tag{2.9}$$

A equação 2.10 define a operação de fechamento de um conjunto f por um elemento estruturante g.

$$fec^g(f) = f\ \ fec\ g = (f \oplus \tilde{g}) \ominus g \tag{2.10}$$

O fechamento em níveis de cinza é uma operação dual à operação de abertura em níveis de cinza. Uma imagem fechada é mais regular e menos rica em detalhes que a imagem original. Os principais efeitos do fechamento sobre uma imagem são: separar padrões escuros

próximos, eliminar os padrões escuros inferiores ao elemento estruturante, manter os padrões claros afastados e conectar os padrões claros mais próximos [14].

2.5.9 Filtros alternados seqüenciais em níveis de cinza

A alternância seqüencial das operações de abertura e fechamento produz poderosos filtros.

Produto de aberturas e de fechamentos

Com a alternância das operações de abertura abe^g e de fechamento fec^g, é possível obter os seguintes filtros: abefec^g, fecabe^g, fecabefec^g e abefecabe^g, como definidos nas equações abaixo.

$$abefec^g(f) = abe^g(fec^g(f))$$

$$fecabe^g(f) = fec^g(abe^g(f))$$

$$fecabefec^g(f) = fec^g(abe^g(fec^g(f)))$$

$$abefecabe^g(f) = abe^g(fec^g(abe^g(f))) \tag{2.11}$$

Este tipo de filtro é muito utilizado antes de operações que aumentem o ruído da imagem, como no caso do gradiente.

Filtros alternados seqüenciais

A equação 2.12 e a 2.13 definem uma família de aberturas e fechamentos de parâmetro ϕ.

$$abe^{\phi}(f) = abe^{\phi g}(f) = abe^{\phi g}(abe^{(\phi-1)g}(...\, abe^g(f))) \tag{2.12}$$

$$fec^{\phi}(f) = fec^{\phi g}(f) = fec^{\phi g}(fec^{(\phi-1)g}(...\, fec^g(f))) \tag{2.13}$$

A partir da definição destas duas famílias, é possível definir filtros alternados seqüencias, como demonstrados nas equações 2.14 e 2.15.

$$abefec^{(i)}(f) = abe^{(i)}(fec^{(i)}(abe^{(i-1)}(fec^{(i-1)}(...abe^{(1)}(fec^{(1)}(f)))))) \tag{2.14}$$

$$fecabe^{(i)}(f) = fec^{(i)}(abe^{(i)}(fec^{(i-1)}(abe^{(i-1)}(...fec^{(1)}(abe^{(1)}(f)))))) \tag{2.15}$$

No processo de filtragem utilizando o filtro alternado seqüencial $abefec^{(i)}$, a primeira iteração fec^1 visa a eliminação de ruídos menores. Na seqüência, a utilização da operação $abe^{(1)}$ visa recuperar a informação prejudicada anteriormente. Utilizando-se elementos estruturantes maiores, é possível eliminar mais ruído. Assim, a aplicação destes filtros objetiva a suavização da imagem, tendo em vista serem excelentes eliminadores de ruídos [14].

2.6 Morfologia em Cores

A cor é um importante descritor e traz consigo um conjunto de informações que não podem ser desconsideradas. Esta valiosa informação existente na cor pode ser utilizada empregado-se a morfologia em cores. Segundo Calixto [9] é possível construir uma relação de ordem em um determinado espaço de cor e com isto, definir uma morfologia para este espaço.

A morfologia matemática define operadores fundamentais em termos de relações de inclusão e ordem. Em imagens binárias ou níveis de cinza, estas relações de ordem possuem aplicações diretas, porém em imagens coloridas, há um grande desafio na ordenação de cores. Segundo Calixto, a dificuldade na definição de uma ordem de cores e as diferenças numéricas entre os seus diversos espaços tornam a definição da Morfologia em Cores uma tarefa complexa.

Nas imagens em tons de cinza, pode-se facilmente aplicar a dilatação ou erosão, pois a dilatação é baseada no máximo e a erosão no mínimo entre os tons de cinza. Mas, no domínio colorido, a determinação do máximo ou mínimo entre cores não é tão trivial.

2.6.1 Ordenações de cores

Considerando-se que a erosão é obtida através do mínimo e que a dilatação é obtida através do máximo nível de cinza entre os pontos envolvidos pelo elemento estruturante, revela-se a necessidade de uma noção de ordem, que é óbvia quando níveis de cinza são utilizados.

Como os tons de cinza variam entre 0 e 255, o mínimo ou o máximo entre tons de cinza é facilmente determinado. Mas, quando uma imagem colorida é considerada, a noção de ordem não é tão simples, pois não existe um consenso do que seria, por exemplo, o mínimo ou máximo entre o verde, o azul e o vermelho.

A maneira mais fácil de se implementar a morfologia em cores é obtendo-se uma ordem marginal em um espaço de cor, isto é, os operadores são aplicados em cada uma das componentes de cor separadamente e depois os resultados são recombinados para gerar a imagem resultante. Porém, esta abordagem possui o problema de gerar cores que não fazem parte da imagem original [9].

Na figura 2.15 observa-se a dilatação utilizando a ordenação marginal no espaço de cores RGB (*red*, *blue*, *green*), na qual emprega-se a dilatação binária em cada uma das componentes da imagem e estas são recombinadas para gerar a imagem dilatada.

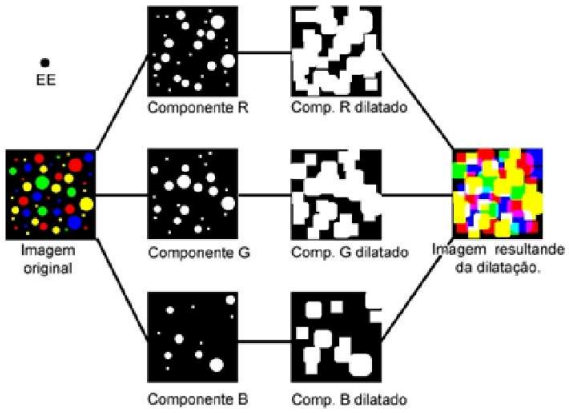

Figura 2.15: Dilatação da imagem colorida usando ordenação marginal [9].

Na figura 2.16 na página seguinte são representados o histograma da imagem original e o histograma do resultado da dilatação utilizando a ordenação marginal. Comparando-se as imagens, pode-se perceber que novas cores que não pertenciam ao conjunto inicial são introduzidas (cores falsas). Porém, este resultado não é esperado em um operador morfológico, pois novas cores podem distorcer o conteúdo das informações, perdendo-se o controle sobre estas mudanças.

Outra técnica da morfologia em cores é a ordenação reduzida, que visa transformar cada vetor representante de uma cor em um escalar, objetivando a aplicação da noção de ordem neste conjunto. Para transformar um dado vetorial em um escalar, considera-se um escalar K como a média de todas as componentes primárias ou como a soma destas componentes. Porém, nesta

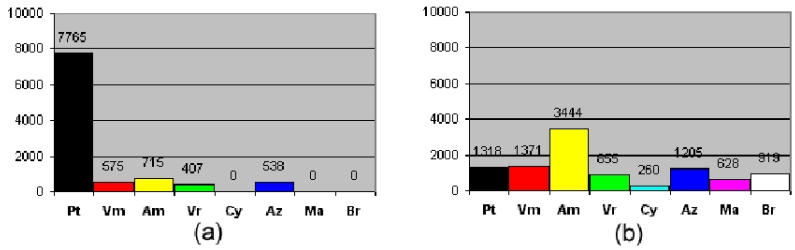

Figura 2.16: (a) Histograma das imagens da figura 2.14 [9].

ordenação não é obrigatório o uso de todas as componentes de cor e poderia-se utilizar o espaço HSV, considerando-se apenas a componente de intensidade V. Desta forma, um vetor h_x, s_x, v_x que representa uma cor x poderia ser reduzido ao escalar v_x [9].

Na ordenação reduzida, não há o surgimento de novas cores, como pode ser exemplificado no histograma da figura 2.17. Neste exemplo, uma imagem foi dilatada por um elemento estruturante plano 11 x 11, utilizando-se o espaço RGB e obtendo-se o escalar K pela soma das componentes primárias da cor. Segundo Calixto, a ordenação reduzida pode empregar diretamente às imagens coloridas, os mesmos operadores morfológicos em tons de cinza.

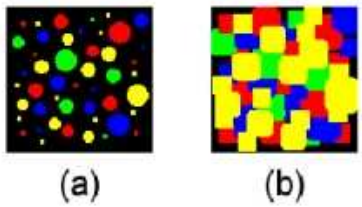

Figura 2.17: (a) Imagem original, (b) Dilatação de (a) utilizando a ordenação reduzida [9].

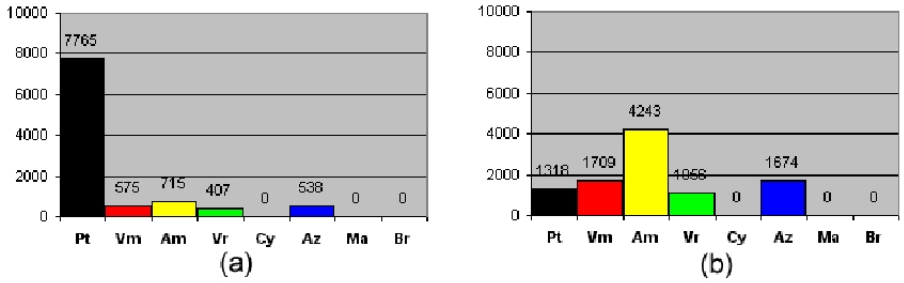

Figura 2.18: Histograma das imagens da figura 2.16 [9].

Outro tipo de ordenação é chamada lexicográfica. Nesta ordenação as cores são, primeiramente, ordenadas por uma componente, depois pela segunda componente e assim por diante.

Na figura 2.19 na página seguinte pode-se observar a dilatação da imagem (a) por um

elemento estruturante plano 11 x 11, utilizando-se a ordem lexicográfica aplicada ao espaço RGB. Neste exemplo, as cores foram ordenadas na ordem de suas componentes, ou seja, primeiramente, ordenadas pela componente R. Depois, as cores com mesmas intensidades em R foram ordenadas pela componente G. Então, as cores com mesmo R e G, foram ordenadas pela componente B. Porém, observa-se que neste tipo de ordenação, a primeira componente ordenada é destacada [9].

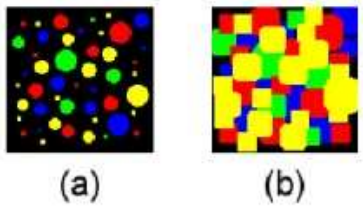

Figura 2.19: (a) Imagem original, (b) Dilatação de (a) utilizando a ordenação lexicográfica [9].

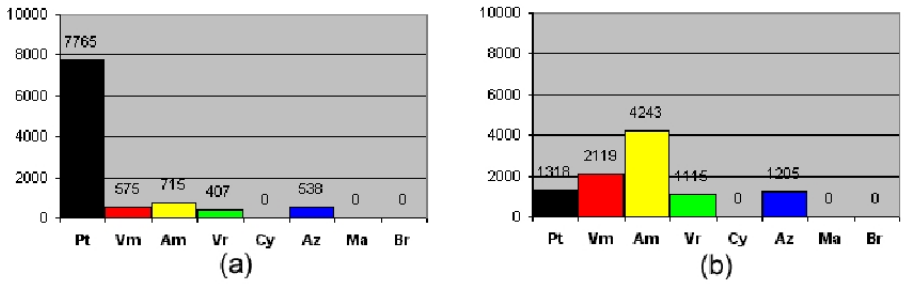

Figura 2.20: Histograma das imagens da figura 2.18 [9].

2.6.2 Formato HSV

O formato HSV representa as cores das imagens através do matiz (*hue*), saturação (*saturation*) e intensidade ou brilho (*value*). Neste espaço, as cores vermelho, amarelo, verde, ciano, azul e magenta ocupam os vértices da base de uma pirâmide hexagonal invertida (figuras 2.21 na próxima página). A altura da pirâmide representa a variação de intensidade. A saturação é diretamente proporcional à distância ao eixo da pirâmide [9].

O matiz é a cor pura dominante e é medida por um ângulo (0°até 360°), começando com o vermelho em 0°, verde com 120°e azul com 240°.

No eixo vertical da pirâmide encontra-se uma escala em tons de cinza . A saturação e a intensidade estão normalizadas variando, assim, entre 0 e 1.

A saturação representa a quantidade do matiz puro acrescentado à cor. O valor 0, no centro da pirâmide, representa nenhum matiz (branco) e o valor 1, na borda da pirâmide, representa uma cor primária pura.

A intensidade é a quantidade de luz presente na cor. A componente de intensidade varia entre 0 e 1, ou seja, o valor 0 representa o preto ou nenhum brilho e o 1 representa a cor brilhante.

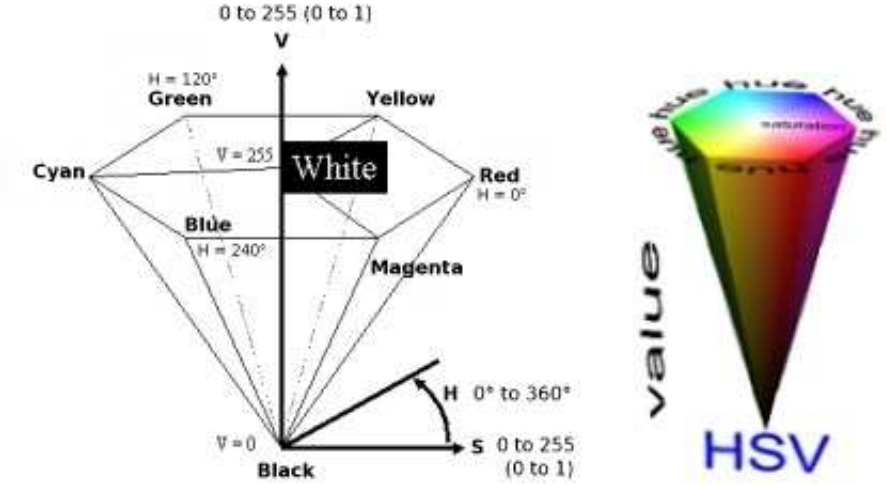

Figura 2.21: Espaço HSV [13].

2.6.3 Ordenação H&S no espaço HSV

Como visto anteriormente, várias ordenações podem ser aplicadas a diferentes espaços de cores. Mas, segundo Calixto, a percepção humana da cor está mais próxima da decomposição da cor em iluminação e cromaticidade. Desta forma, a iluminação pode ser processada utilizando os operadores da morfologia em níveis de cinza, haja vista a correspondência entre variações de intensidade de iluminação e níveis de cinza. E, ainda, a informação extra da imagem colorida pode ser encontrada na cromaticidade.

A abordagem de Calixto [9] define uma nova relação de ordem no espaço de cor HSV e determina a morfologia para este espaço. A sua técnica de ordenação é baseada na característica deste espaço em reunir a informação cromática nas componentes H e S. Esta abordagem utiliza a ordenação reduzida e lexicográfica.

Nesta técnica, a cor é ordenada por meio da constante de cromaticidade e depois pela iluminação. *A constante de cromaticidade é um valor escalar obtido pela redução das compo-*

nentes de cromaticidade, matiz e saturação (H e S), em um único escalar [9].

O matiz é uma grandeza angular que vai de 0°à 360°e a saturação é um gradiente em cinza variando de 0 à 255. Como o matiz e a saturação são grandezas diferentes, para combiná-las é necessário normalizar os valores para obter grandezas equivalentes. Assim, a técnica define que o matiz e a saturação devem variar entre 0 e 1.

Mesmo normalizados, o matiz e a saturação são conceitualmente diferentes, tendo em vista que a saturação é representada em uma reta e o matiz é representado na circunferência do círculo trigonométrico.

Devido a esta diferença de representação, os valores 0 e 1 representam saturações opostas, mas representam o mesmo valor de matiz, pois 0°e 360°são equivalentes. Por este motivo, foi definida a distância de matiz, $dH(h_a, h_b)$, que é o menor ângulo entre dois matizes na circunferência do círculo trigonométrico. Assim, a maior distância entre duas cores é 180°.

Definição 2.11. Seja C a circunferência de um círculo trigonométrico e h_a, $h_b \in C$ dois valores de matiz, define-se a distância de matiz pela equação 2.16:

$$dH(h_a, h_b) = \begin{cases} \frac{|h_a - h_b|}{180^\circ}, & se\ |h_a - h_b| \leq 180^\circ \\ \frac{360^\circ - |h_a - h_b|}{180^\circ}, & se\ |h_a - h_b| > 180^\circ \end{cases} \tag{2.16}$$

Nesta técnica, para se ordenar o matiz, primeiramente, define-se um valor inicial como mínimo, calcula-se a distância de um matiz qualquer ao mínimo e, depois, ordena-se segundo essa distância.

Com o matiz e saturação representados em uma mesma escala, Calixto definiu a função ”constante de cromaticidade” para transformar as duas componentes de cromaticidade em apenas um escalar.

Definição 2.12. Seja a e b duas cores pertencentes ao espaço HSV, sendo $a = (h_{a,} s_a, v_a)$ e $b = (h_b, s_b, v_b)$. A função constante de cromaticidade é definida na equação 2.17:

$$c(a, b) = max(|s_a - s_b|, dH(h_a, h_b)) \tag{2.17}$$

Dado um conjunto de cores e utilizando-se a função constante de cromaticidade, é possível determinar a cor mais próxima como sendo aquela que possui a menor constante de cromaticidade.

2.6.4 Erosão e dilatação coloridas

Definida a ordenação no espaço de cores, é possível obter o máximo e mínimo em um conjunto de cores. Sejam $a, b \in HSV$, o máximo colorido entre a e b é definido pela equação 2.18 e o mínimo colorido entre a e b é definido pela equação 2.19:

$$a \veebar b = max\{c(a,o), c(b,o)\} \tag{2.18}$$

$$a \barwedge b = min\{c(a,o), c(b,o)\} \tag{2.19}$$

sendo o a cor eleita como a menor do espaço.

Assim, pode-se definir a erosão e dilatação colorida para o espaço HSV utilizando-se a constante de cromaticidade. A erosão colorida, pelo elemento estruturante B, em um ponto x de uma imagem f é definida pela equação 2.20:

$$\varepsilon_B(f)(x) = \barwedge\{f(y) : y \in D_{B_x}\} \tag{2.20}$$

onde $\barwedge$ indica o mínimo colorido entre duas cores.

A dilatação da imagem f pelo elemento estruturante B é definida pela equação 2.21:

$$\delta_B(f)(x) = \veebar\{f(y) : y \in D_{B_x}\} \tag{2.21}$$

onde $\veebar$ indica o máximo colorido entre duas cores.

O elemento estruturante utilizado é plano, pois apenas define uma vizinhança de influência para a escolha do máximo ou mínimo definidos.

Com as operações básicas (dilatação e erosão) definidas no espaço, todas as outras operações podem ser obtidas através da combinação destas, como nas binárias ou cinzas.

A escolha da cor mínima é um importante fator que influencia nas operações de erosão e dilatação colorida. Nos exemplos das figuras 2.22 e 2.24 na página seguinte, pode-se verificar como a cor mínima influenciou a dilatação colorida da imagem. Nestes exemplos, os resultados da dilatação propagaram as cores mais distantes da cor mínima.

Uma das vantagens desta morfologia em cores proposta por Calixto é que, com a definição da cor mínima como sendo a cor do fundo, evita-se a inversão entre erosão e dilatação. Esta inversão pode ser verificada no exemplo da figura 2.26 na página 29, em que foi utilizada

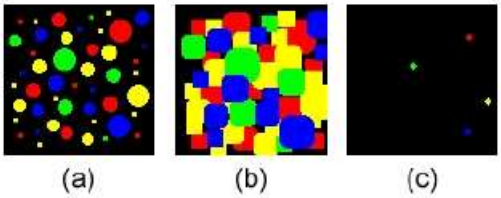

Figura 2.22: (a) Imagem original, (b) Dilatação usando o matiz vermelho como cor mínima, (c) Erosão usando o matiz vermelho como cor mínima [9].

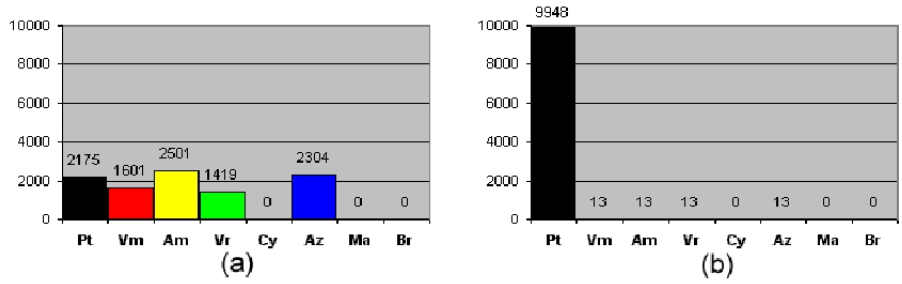

Figura 2.23: (a) Histograma da imagem dilatada, (b) Histograma da imagem erodida [9].

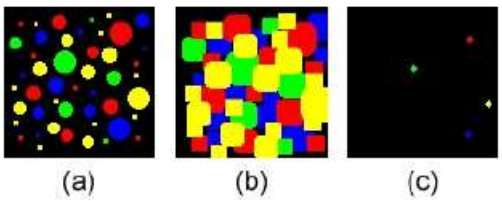

Figura 2.24: (a) Imagem original, (b) Dilatação usando o matiz azul como cor mínima, (c) Erosão usando o matiz azul como cor mínima [9].

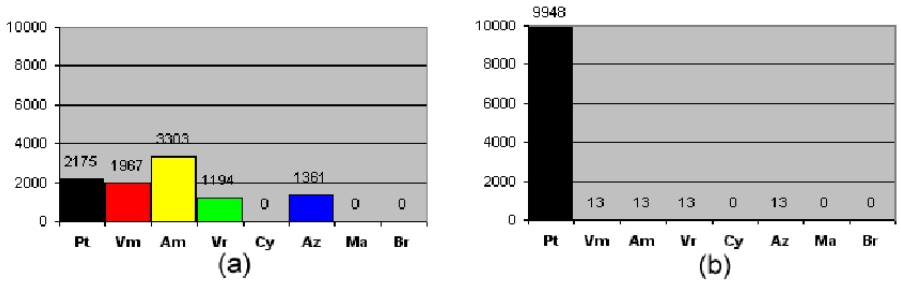

Figura 2.25: (a) Histograma da imagem dilatada, (b) Histograma da imagem erodida [9].

uma cor mínima ”maior” do que as cores dos objetos, e com isto, a dilatação da imagem significou a dilatação do fundo, causando a erosão dos objetos.

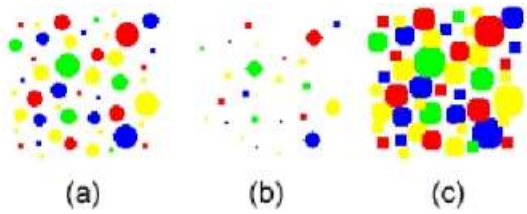

Figura 2.26: Exemplo de inversão dos operadores: (a) Imagem original, (b) Dilatação, (c) Erosão [9].

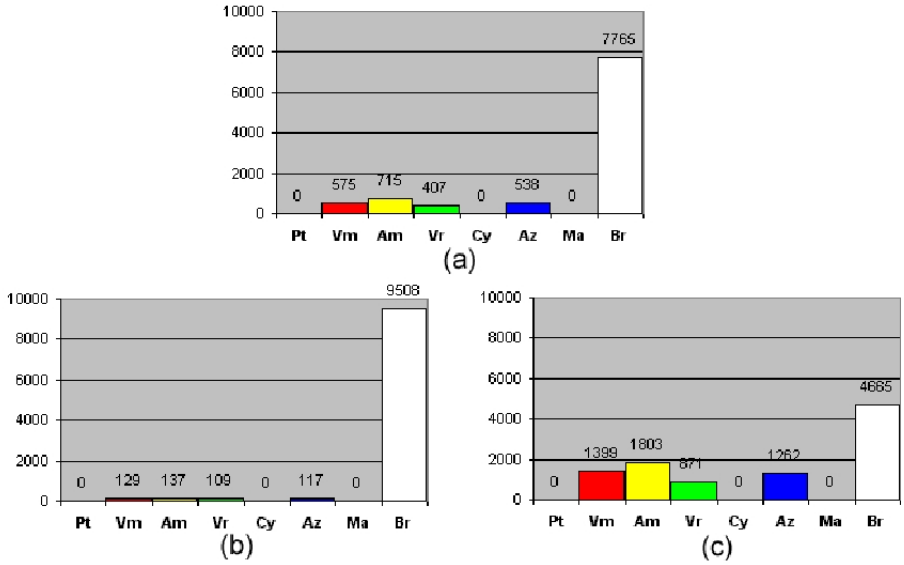

Figura 2.27: (a) Histograma da imagem original, (b) Histograma da imagem dilatada, (c) Histograma da imagem erodida [9].

Quando a métrica proposta por Calixto é utilizada, os operadores mantém-se coerentes, pois a cor mínima utilizada é a cor de fundo. A aplicação desta técnica pode ser exemplificada na figura 2.28 na página seguinte, em que a cor mínima definida foi o ciano e na figura 2.29 na próxima página, em que a cor mínima utilizada foi o verde.

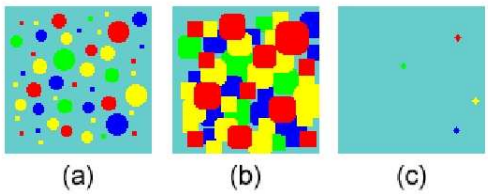

Figura 2.28: Cor mínima utilizada igual a cor do fundo: (a) Imagem original, (b) Dilatação, (c) Erosão [9].

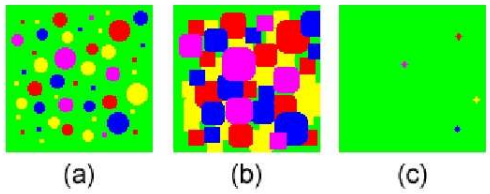

Figura 2.29: Cor mínima utilizada igual a cor do fundo: (a) Imagem original, (b) Dilatação, (c) Erosão [9].

2.7 Limiarização

2.7.1 Introdução

A limiarização é uma das mais simples técnicas utilizadas para a segmentação de imagens. Também chamada de binarização, a limiarização visa segmentar imagens em regiões de interesse, descartando as regiões não desejadas [34][27].

As técnicas de limiarização mais simples tentam utilizar apenas um limiar para segmentar os objetos de interesse de uma imagem. Contudo, é difícil obter um único limiar que realize uma segmentação satisfatória para toda a imagem. Nestes casos, são necessárias técnicas de limiarizações variáveis e multiníveis que utilizam medidas estatísticas ou realizar a segmentação de pontos da imagem por limiares diferentes [25].

2.7.2 Limiarização Global

A limiarização global visa segmentar imagens, separando os objetos de interesse do fundo ao qual pertencem. Se uma imagem possui objetos escuros e um fundo claro, os *pixels* do fundo possuirão níveis de cinza mais altos que os objetos da imagem, pois a cor branca é representada pelo nível de cinza 255 e a preta é representada por 0. A figura 2.30 na página seguinte, exemplifica um histograma de distribuição de níveis de cinza e a sua imagem original [25].

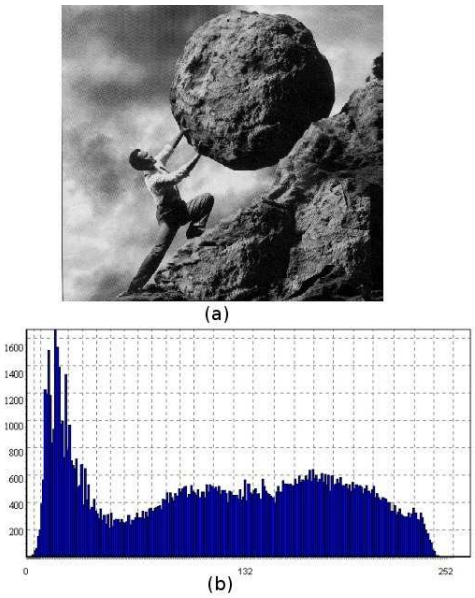

Figura 2.30: (a) Imagem original em níves de cinza, (b) Histograma de distribuição de níveis de cinza [25].

Diante do exposto, seria possível separar os *pixels* de uma imagem em dois grupos, baseados em seus valores de níveis de cinza, permitindo a distinção entre fundo e objetos. Contudo, existem situações em que os objetos da imagem possuem bordas suavizadas e nestes casos, não é possível a determinação de dois níveis de cinza que caracterizem fundo e objetos. Em casos como estes, a presença de ruído na imagem pode agravar ainda mais a situação.

Para resolver este problema, pode-se determinar um nível de cinza T (entre os dois níveis de cinza dominantes) que representa um limiar diferenciador para as classes objetos e fundo [27]. Este exemplo pode ser verificado na figura 2.31 na próxima página, cujo limiar T é escolhido por um método não paramétrico e não supervisionado descrito em Otsu [31]. A partir deste limiar, pode-se obter uma imagem binária cujos objetos são pretos e o restante é branco.

Se $f(x,y)$ é a imagem original, o produto da limiarização é obtido testando-se a imagem original, *pixel* a *pixel*, contra o limiar determinado. Se $f(x,y) > T$, então o *pixel* é considerado fundo, caso contrário, o *pixel* é considerado objeto, como definido na equação 2.22, onde $b(x,y)$ é o limiar binário [25].

$$b(x,y) = \begin{cases} 255, & se\ f(x,y) > T \\ 0, & se\ f(x,y) \leq T \end{cases} \tag{2.22}$$

Em situações ideais, verifica-se a existência de um vale profundo, entre dois cumes no histograma da imagem, que corresponde aos objetos e fundo. Nestes casos, a limiarização glo-

bal produz resultados satisfatórios, pois o limiar ideal é determinado no fundo do vale. Porém, na maioria dos casos reais, não é tão trivial determinar fundos de vales [25].

A seleção de um único limiar possui algumas desvantagens como exposto por O'Gorman e Mattana [30]. Quando há falta de contraste entre objetos e fundo, ou a presença de ruídos e imagens com objetos escassos, ou ainda, o objeto possui partes mais claras que o fundo, o cume que representa os níveis de cinza dos objetos é muito menor que o do fundo. Apesar de suas desvantagens, as técnicas de limiarizações globais são muito utilizadas por sua rapidez de processamento.

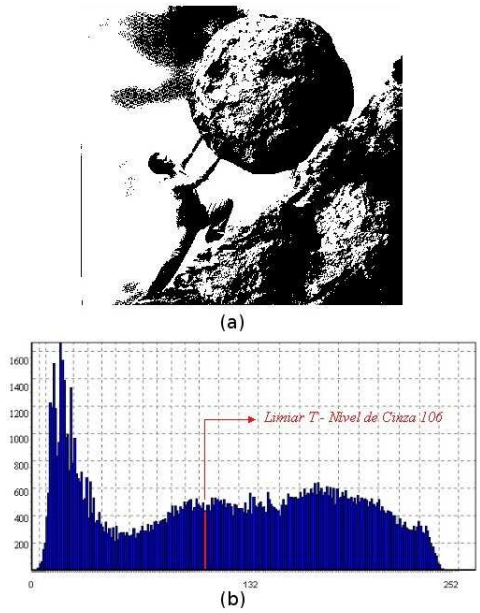

Figura 2.31: (a) Imagem binária produzida pela limiarização de Otsu no nível de cinza 106, (b) Histograma de distribuição de níveis de cinza da imagem original [25].

Limiarização por Anisotropia

A Limiarização por Anisotropia de Pun [35] propõe uma avaliação de um limiar ótimo $t*$ baseado no conhecimento a posteriori de entropia. Um coeficiente de anisotropia α é estabelecido como na equação 2.23.

$$\alpha = \frac{\sum_{i=0}^{m} p_i log_e p_i}{\sum_{i=0}^{L-i} p_i log_e p_i} \tag{2.23}$$

onde i representa o nível de cinza ($0 \leq i \leq L - i$), $L - i$ se refere ao número máximo de níveis de cinza, p_i indica a probabilidade do nível de cinza i e m é o menor inteiro verificado na equação 2.24.

$$\sum_{i=0}^{m} p_i \geq 0.5 \tag{2.24}$$

Sendo assim, o valor do limiar ótimo t* é tal como definido na equação 2.25.

$$\sum_{i=0}^{t*} p_i = \begin{cases} 1-\alpha & se\ \alpha \leq 0.5 \\ \alpha & se\ \alpha > 0.5 \end{cases} \tag{2.25}$$

De acordo com Kapur [18] este algoritmo introduz um viés, pois sempre fornece um valor limiar superior ou igual a *m*.

2.7.3 Limiarização Adaptativa

Como exposto acima, a limiarização global produz resultados satisfatórios quando o histograma de distribuição de níveis de cinza possui picos distintos e separados, representando objetos e fundo. Assim sendo, em situações diferentes da ideal, um limiar local deve ser utilizado.

A limiarização adaptativa local pode fornecer melhores resultados para imagens em que o histograma não possui picos bem definidos. Neste tipo de limiarização, um limiar individual é determinado para cada *pixel*, definido a partir do alcance de intensidade estimado na vizinhança local [27] .

Nas técnicas adaptativas, definidas pela equação 2.26, é necessário dividir-se a imagem original em imagens menores, determinando um limiar para cada sub-imagem. Se um limiar não puder ser definido para alguma das imagens menores, este pode ser interpolado a partir dos limiares das sub-imagens vizinhas. Por fim, cada imagem menor é processada utilizando seu limiar local [25].

$$T = T[x, y, p(x, y), f(x, y)] \tag{2.26}$$

onde $f(x, y)$ é o nível de cinza do ponto (x, y) na imagem original, e $p(x, y)$ é uma propriedade local deste ponto, descrita a seguir.

Diferentemente da limiarização global, o limiar T não depende apenas do nível de cinza do ponto. A propriedade do ponto, definida pelo fator $p(x, y)$, é um dos mais importantes fatores no cálculo do limiar [Milstein (1998)]. O cálculo deste fator é normalmente baseado no ambiente em que o ponto está inserido, para levar em consideração a influência de ruído e

iluminação. A seleção do tamanho da janela que definirá a vizinhança é o grande desafio para as técnicas de limiarização adaptativa, pois exige o prévio conhecimento da imagem [34] [25].

A limiarização adaptativa produz resultados satisfatórios para imagens com histogramas bimodais e quando os objetos forem relativamente pequenos e não muito próximos uns dos outros [34] [25].

Limiarização Adaptativa de Bernsen

Como visto anteriormente, a maior dificuldade no uso da limiarização global é a determinação de um único limiar global para a segmentação da imagem. As partes mais escuras da imagem são determinadas como preto e as partes mais claras como branco. Neste sentido, é indispensável a utilização de um método adaptativo, cujo limiar é calculado para cada *pixel* baseado em sua vizinhança.

A técnica de limiarização adaptativa de Bernsen, define para cada *pixel* (x, y), um limiar $T(x, y)$ como definido na equação 2.27.

$$T(x, y) = \frac{(Pmenor + Pmaior)}{2} \tag{2.27}$$

onde *Pmenor* e *Pmaior* são os mais baixos e mais altos valores de *pixel* em níveis de cinza em uma vizinha de *RxR* quadrada e de centro em (x, y).

Porém, se a medida de contraste $C(x, y) = (Pmaior - Pmenor)$ for menor que L, que é o contraste mínimo, então a vizinhança só consiste em uma classe, preto ou branco. Os valores de R e L sugeridos são 15 [4].

2.7.4 Limiarização Multinível

Embora as técnicas de limiarização em dois níveis sejam simples, em alguns casos, as imagens possuem histogramas de distribuição de níveis de cinza que não são bimodais e, conseqüentemente, as técnicas de limiarização em dois níveis não apresentam resultados satisfatórios. Desta forma, quando uma imagem possui vários objetos que se diferenciam do fundo, o seu histograma de distribuição de níveis de cinza será multimodal. Nestes casos, pode-se utilizar a limiarização multinível, cujo limiar é determinado pela localização dos vales que separam os objetos [36].

A limiarização multinível possibilita a segmentação de imagens em várias classes. Se uma imagem possui um histograma de distribuição de níveis de cinza com três cumes, con-

seqüentemente esta imagem poderá ser segmentada usando-se dois limiares, como pode ser verificado na figura 2.32. Estes limiares dividem o conjunto de valores em três intervalos distintos.

Os métodos de limiarização multinível devem segmentar uma imagem para os diferentes objetos com propriedades similares. Fatores como a distribuição dos níveis de cinza, pequenos objetos e a sobreposição de objetos podem interferir na qualidade da segmentação [10]. Assim sendo, a obtenção de múltiplos limiares que realizem uma segmentação satisfatória das regiões de interesse não é uma tarefa trivial [42].

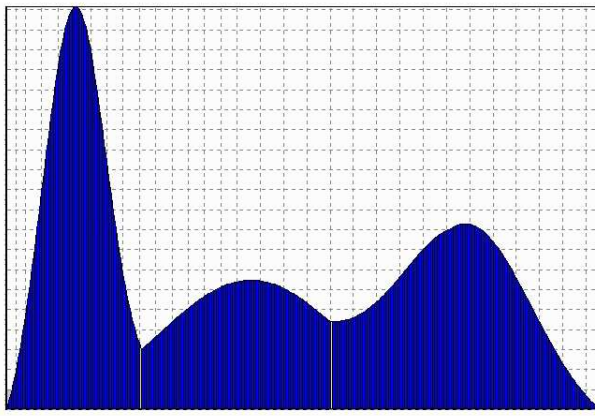

Figura 2.32: Histograma de distribuição de níveis de cinza multimodal com três cumes e dois limiares [25].

Capítulo 3

Estado da Arte

Como será visto neste capítulo, há uma grande variedade de abordagens para a segmentação temporal de vídeo e, conhecendo-se algumas de suas limitações, é possível obter um direcionamento para o desenvolvimento de novas técnicas.

3.1 Abordagens para segmentação de vídeo sem compressão

Existem muitas abordagens para a segmentação de vídeo em domínio sem compressão. Nesta seção será feita uma revisão das principais técnicas descritas na literatura.

A maioria das técnicas de segmentação de vídeo utiliza medidas de dissimilaridade de características como cor ou forma para detecção de transições entre quadros. Um corte pode ser detectado se a medida de dissimilaridade entre dois quadros sucessivos for suficientemente grande.

3.1.1 Comparação *pixel* a *pixel* (*Template Matching*)

A técnica de comparação *pixel* a *pixel* avalia a diferença de intensidade ou cor dos *pixels* correspondentes em dois quadros consecutivos. A soma da diferença absoluta de *pixels* de quadros sucessivos é calculada e comparada a um limiar [20] [22].

A soma da diferença absoluta de *pixels* para imagens em níveis de cinza e imagens coloridas são definidas, respectivamente, pelas equações 3.1 e 3.2:

$$D(i, i+1) = \frac{\sum_{x=1}^{X} \sum_{y=1}^{Y} |P_i(x,y) - P_{i+1}(x,y)|}{XY} \tag{3.1}$$

$$D(i, i+1) = \frac{\sum_{x=1}^{X}\sum_{y=1}^{Y}\sum_{c}|P_i(x,y,c) - P_{i+1}(x,y,c)|}{XY} \tag{3.2}$$

onde i e $i+1$ são dois quadros consecutivos de tamanho X x Y, $P_i(x,y)$ é a intensidade do *pixel* de coordenadas (x,y) no quadro i, c é o índice para as componentes de cores e $P_i(x,y,c)$ é a componente de cor do *pixel* (x,y) no quadro i.

Um corte é detectado se a diferença $D(i, i+i)$ é superior a um limiar pré-determinado. A principal desvantagem de métodos baseados na comparação de *pixels* é que eles são sensíveis a movimentos de objetos e câmeras. Desta forma, cortes podem ser detectados erroneamente quando um objeto de uma pequena parte de um quadro sofre uma mudança rápida e grande.

3.1.2 Comparação baseada em bloco

A comparação baseada em bloco utiliza características locais da imagem, visando melhorar a sensibilidade ao movimento de câmera e objetos existente na técnica de comparação *pixel* a *pixel*.

Cada quadro i é dividido em b blocos que são comparados a seus blocos correspondentes no quadro consecutivo $i+1$. A diferença entre blocos de dois quadros sucessivos é calculada pela equação 3.3 [22].

$$D(i, i+1) = \sum_{k=1}^{b} c_k DP(i, i+1, k) \tag{3.3}$$

onde c_k é um coeficiente pré-determinado para o bloco k e $DP(i, i+1, k)$ é uma combinação parcial entre os blocos nos quadros i e $i+1$.

Kasturi e Jain [19] mostram que a semelhança de blocos pode ser obtida calculando-se a taxa de probabilidade nos blocos correspondentes como na equação 3.4:

$$\lambda_k = \frac{\left[\frac{\sigma_{k,i}+\sigma_{k,i+1}}{2} + \left(\frac{\mu_{k,i+1}-\mu_{k,i+1}}{2}\right)^2\right]^2}{\sigma_{k,i}\sigma_{k,i+1}} \tag{3.4}$$

onde $\mu_{k,i}$ e $\mu_{k,i+1}$ são a média dos valores de intensidade de dois blocos k correspondentes nos quadros i e $i+1$, e $\sigma_{k,i}$, $\sigma_{k,i+1}$ são as variâncias destes quadros sucessivos. Desta forma, são contados apenas o número de blocos para o qual a probabilidade de mudança é maior que o

limiar T_1, como na equação 3.5:

$$DP(i, i+1, k) = \begin{cases} 1 & se\ \lambda_k > T_1 \\ 0 & para\ os\ outros\ casos \end{cases} \quad (3.5)$$

Um corte é detectado quando o número de blocos alterados $DP(i, i+1, k)$ é maior que um limiar T_2 e $c_k = 1$ para todo k.

Este método é mais tolerante a movimentos de pequenos objetos se comparado ao método de comparação *pixel* a *pixel*. Porém, se dois blocos diferentes possuírem a mesma função de densidade, mudanças não são detectadas. Outra desvantagem desta técnica é a sua demora de processamento devido à complexidade das fórmulas estatísticas usadas.

O método chamado *net comparison*, proposto por Xiong et al. [43], avalia apenas partes da imagem. Janelas são comparadas usando-se a diferença entre a média de valores de níveis de cinza ou cores. Se a diferença é maior que um limiar, então pode-se considerar que a região mudou. Um corte é detectado se o número de janelas alteradas é maior que outro limiar [22].

3.1.3 Comparação de histogramas

Comparação global de histogramas

A abordagem da comparação de histogramas é baseada na premissa de que dois quadros que não possuem grandes mudanças, possuem pequena diferença nos histogramas. Histogramas são invariantes a rotação de imagem para baixa variação de ângulo e escala [22].

Com a comparação de histogramas de imagens sucessivas pode-se obter a redução da sensibilidade a movimentos de câmera e objetos. A desvantagem desta abordagem é que duas imagens com histogramas similares podem ter conteúdos completamente diferentes.

A comparação de histogramas de níveis de cinza é utilizada na abordagem de Nagasaka e Tanaka [28]. Um corte é detectado se a soma da diferença absoluta de histogramas entre dois quadros consecutivos $D(i, i+1)$ é maior que um limiar T, como pode ser visto na equação 3.6:

$$D(i, i+1) = \sum_{j=1}^{n} |H_i(j) - H_{i+1}(j)| \quad (3.6)$$

onde $H_i(j)$ é valor do histograma para o nível de cinza j no quadro i, n é o total de níveis de cinza e j é o nível de cinza.

A abordagem de Zhang e Smoliar [46] compara histogramas de cores ao invés de níveis de cinza. A equação 3.6 é usada e j passa a ser o valor do código da intensidade das três cores

de um *pixel*.

Algumas abordagens [28] propõem o uso do teste do χ^2 para comparar os histogramas de cores $H_i(j)$ e $H_{i+1}(j)$ de dois quadros consecutivos i e $i+1$, como na equação 3.7. Um corte é detectado se a diferença $D(i, i+1)$ é maior que o limiar T. Esta abordagem possui a desvantagem de reforçar a diferença de movimentos de objetos e câmera entre dois quadros sucessivos e de ter maior custo computacional devido aos testes χ^2 realizados [22].

$$D(i, i+1) = \sum_{j=1}^{n} \frac{|H_i(j) - H_{i+1}(j)|^2}{H_{i+1}(j)} \tag{3.7}$$

O método *twin-comparison* [46] avalia a diferença acumulativa entre os quadros na transição gradual. Cortes são detectados se a diferença entre quadros consecutivos é maior que um limiar alto. O quadro de início de uma transição gradual pode ser detectado quando a diferença entre quadros consecutivos ultrapassa um limiar pequeno. O final de uma transição é detectado quando a diferença entre quadros consecutivos é menor que o limiar e a diferença acumulada é maior que o limiar alto, como pode ser observado na figura 3.1 [22].

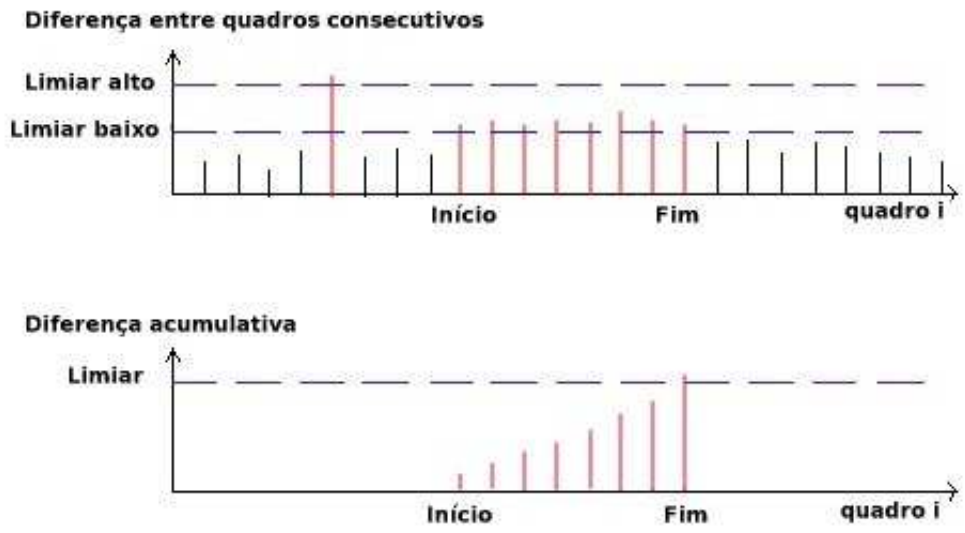

Figura 3.1: *Twin comparison*: diferença de histogramas entre quadros consecutivos e diferença acumulada

Variações desta abordagem consideram dados estatísticos para detectar uma transição. Outros espaços de cores como RGB (*Red, Green, Blue*), HSI (*Hue, Saturation, Intensity*), HSV (*Hue, Saturation, Value*) também são utilizados.

Comparação local de histograma

A comparação local histograma mistura a abordagem baseada na comparação de blocos e a abordagem baseada em histogramas, visando reduzir a sensibilidade a movimentos de objetos e câmera, utilizando informação espacial para resultados mais precisos [22].

A diferença entre os valores de histogramas de níveis de cinza entre os quadros i e $i + 1$ é calculada como nas equações 3.8 e 3.9:

$$D(i, i + 1) = \sum_{k=1}^{b} DP(i, i + 1, k) \tag{3.8}$$

$$DP(i, i + 1, k) = \sum_{j}^{n} |H_i(j, k) - H_{i+1}(j, k)| \tag{3.9}$$

onde $H_i(j, k)$ é o valor do histograma para o nível de cinza j do bloco k e b é o total de blocos.

Nagasaka e Tanaka [28] compara estatísticas utilizando diferenças de nível de cinza, cor de *pixels* e comparações de histogramas. Melhores resultados podem ser obtidos dividindo-se a imagem em 16 regiões e usando o teste χ^2 nas regiões para descartar as maiores diferenças provenientes dos efeitos de movimentos de objetos e câmera.

Existe, ainda, abordagens que sugerem que blocos das imagens sejam comparados usando-se histogramas no espaço de cor RGB ou a comparação de histogramas no espaço HSV (matiz, saturação e valor), para diminuir a diferença quadro a quadro causada pela mudança de intensidade ou sombra.

3.1.4 Segmentação temporal de vídeo baseada em agrupamento

A técnica chamada segmentação temporal de vídeo por agrupamento (*clustering*) [32] define a existência de duas classes: uma onde há mudança de cena e outra onde não há mudança de cena. O algoritmo K-*means* é usado para agrupar diferenças entre quadros. Os quadros que estão agrupados no *cluster* de mudança de cena, que são temporariamente adjacentes, são classificados como pertencentes à transição gradual e os outros quadros deste *cluster* são classificados como cortes.

A medida da diferença de histogramas para quadros consecutivos (equação 3.6), no espaço de cores RGB ou YUV , e o teste estatístico χ^2 são usados. O teste χ^2 é usado para se detectar o número de transições corretas.

Uma desvantagem desta abordagem é que ela não é capaz de reconhecer algumas

transições graduais, porém esta técnica elimina a necessidade de utilização de limiares e permite que várias características sejam usadas simultaneamente para melhorar a performance [22].

3.1.5 Segmentação temporal de vídeo baseada em característica

A abordagem de segmentação temporal baseada em característica [45] utiliza a análise de intensidade de bordas entre quadros consecutivos. Quando cortes ou *dissolves* ocorrem, diferentes intensidades de bordas surgem longe do local das bordas antigas, e estas desaparecem do local das novas bordas. Desta forma, transições podem ser detectadas pela contagem dos *pixels* que apareceram e desapareceram das bordas.

Um algoritmo de compensação de movimento pode ser usado, para casos de movimento de objetos e câmera. Com ele, estima-se o movimento global entre quadros, que é usado para alinhar os quadros antes de detectar os *pixels* de borda que apareceram e desapareceram. Porém, esta técnica não é capaz de lidar com vários objetos se movendo rapidamente e acusa falsas transições (falsos positivos) devido as limitações dos métodos de detecção de bordas [22].

3.1.6 Segmentação temporal de vídeo dirigida pelo modelo

Na segmentação temporal dirigida pelo modelo, é possível a utilização de abordagens *bottom-up* que tratam o problema do ponto de vista da análise de dados, ou a utilização de algoritmos *top-down* baseados em modelos matemáticos de dados de vídeo [22].

A técnica apresentada por Hampapur et al. [17] utiliza um algoritmo que realiza a identificação dos limites de tomadas baseado no modelo matemático do processo de produção de vídeo, usado como base para a classificação de tipos de edições de vídeo (cortes, *fades* e *dissolves*).

Fades e *dissolves* são edições cromáticas e podem ser modeladas pela equação 3.10:

$$S(x,y,t) = S_1(x,y,t)(1 - \frac{t}{l_1}) + S_2(x,y,t)(1 - \frac{t}{l_2}) \tag{3.10}$$

onde $S_1(x,y,t)$ e $S_2(x,y,t)$ são duas tomadas que estão sendo editadas, $S(x,y,t)$ é a tomada editada e l_1, l_2 são o número de quadros para cada tomada durante a edição.

As classificações baseadas em modelos identificam características de diferentes classes de tomadas. Vetores de características são alimentados para classificação de quadros e segmentação temporal de vídeo. Esta abordagem é sensível a movimentos de câmera e objetos.

Existem ainda outras técnicas de segmentação dirigida pelo modelo, permitindo detectar transições através do modelo de mudança de intensidade durante certas transições, ou através do uso do modelo de Markov escondido (HMM). O HMM permite que características sejam incluídas no vetor de características.

3.2 Abordagens para segmentação de vídeo comprimido

A maioria das abordagens de segmentação de vídeo comprimido utiliza termos DC para a construção de imagens DC (versões reduzidas do quadro real, seção 2.4), e sobre as imagens DC são aplicadas técnicas como a soma da diferença absoluta entre *pixels* e a comparação de histogramas. Outras abordagens utilizam a comparação e cálculos sobre os coeficientes DCT entre quadros.

Segundo Koprinska e Carrato [22], a grande vantagem da segmentação de vídeo em domínio com compressão é a possibilidade do uso de informações pré-computadas que estão disponíveis no *stream* de vídeo comprimido. Outros aspectos positivos são a rapidez nas operações devido a taxas de dados mais baixas, redução na complexidade computacional e tempo poupado, pois não é necessário aplicar a descompressão.

3.2.1 Segmentação de vídeo temporal baseada em coeficientes DCT

A abordagem de Arman et al. [3] sugere a detecção de cortes baseada em coeficientes DCT de quadros I. Desta forma, um subconjunto de coeficientes DCT de um subconjunto de blocos é determinado para cada quadro, visando a construção de um vetor $V_i = \{c_1, c_2, c_3, ...\}$, que representa o quadro i da seqüência de vídeo no espaço DCT. O produto interno normalizado é usado para encontrar a diferença entre quadros i e $i + \varphi$, como pode ser verificado na equação 3.11:

$$D(i, i+\varphi) = \frac{V_i . V_{i+\varphi}}{|V_i| \left|V_{i+\varphi}\right|} \tag{3.11}$$

Um corte é detectado se $1 - |D(i, i+\varphi)| > T_1$, sendo T_1 um limiar. Um segundo limiar T_2 ($0 < T_1 < T_2 < 1$) é utilizado para avaliar o corte, visando reduzir falsos positivos advindos da movimentação de câmera e objeto. Se $T_1 < 1 - |D(i, i+\varphi)| < T_2$, os dois quadros são descomprimidos e examinados pela comparação de seus histogramas de cor [22].

A abordagem de Zhang et al. [48] utiliza a comparação *pixel* a *pixel* para o coeficiente DCT de blocos de quadros do vídeo. A diferença do bloco l de dois quadros que estão a φ quadros pode ser verificada na equação 3.12:

$$DP(i, i+\varphi, l) = \frac{1}{64}\sum_{k=1}^{64}\frac{\left|c_{l,k}(i) - c_{l,k}(i+\varphi)\right|}{max\left[c_{,k}(i), c_{l,k}(i+\varphi)\right]} > T_1 \tag{3.12}$$

onde $c_{l,k}(i)$ é o coeficiente DCT do bloco l no quadro i, $k = 1, ..., 64$ e l depende do tamanho do quadro.

Se a diferença é maior que um limiar T_1, considera-se que o bloco l mudou. Se o número de blocos alterados é maior que um segundo limiar T_2, uma transição entre dois quadros é detectada. Esta abordagem requer menor custo computacional que a abordagem anterior.

Para reduzir o tempo de processamento, os algoritmos poderiam ser aplicados apenas a quadros I de vídeo comprimido MPEG, porém a resolução temporal poderia ser diminuída. Uma desvantagem destes métodos é a incapacidade de lidar com a transição gradual ou falsos positivos introduzidos pelo movimento de câmera e objetos [22].

3.2.2 Segmentação temporal de vídeo baseada em termos DC

A abordagem de Yeo e Liu [44] cria e compara as imagens DC (versões espacialmente reduzidas dos quadros). Imagens DC são construídas a partir de termos DC, que representam a média do bloco. Os termos DC de quadros I estão diretamente disponíveis no *stream* MPEG, mas os quadros B e P são estimados usando vetores de movimentos e os coeficientes DCT de quadros I anteriores.

Quando as técnicas utilizam a diferença de *pixels* sobre quadros inteiros, os resultados são prejudicados pelos movimentos de câmera e objetos. Mas, quando são utilizadas métricas baseadas na diferença de *pixels* sobre imagens DC, os resultados fornecidos são satisfatórios, porém computacionalmente mais caros [22].

Como em abordagens de comparação *pixel* a *pixel*, as transições abruptas são detectadas utilizando-se a medida de similaridade, baseada na soma das diferenças absolutas de pixel de duas imagens DC consecutivas como na equação 3.13:

$$D(l, l+1) = \sum_{i,j}(|P_l(i, j) - P_{l+1}(i, j)|) \tag{3.13}$$

onde l e $l + 1$ são duas imagens DC consecutivas e $P_l(i, j)$ é o valor de intensidade do *pixel* na

imagem DC de coordenadas (i, j).

Yeo e Liu sugerem a utilização de limiares locais e uma janela deslizante, visando examinar m diferenças de quadros consecutivos. Um corte entre quadros l e $l+1$ é detectado se: $D(l, l+1)$ é o máximo dentro de uma janela deslizante simétrica de tamanho $2m-1$ e se $D(l, l+1)$ é n vezes a segunda maior da janela. As transições graduais são detectadas comparando-se cada quadro com o seguinte *k-ésimo* quadro, sendo k maior que o número de quadros na transição gradual. As transições graduais g_n são determinadas como na equação 3.14, na forma de transição linear de c_1 para c_2 no intervalo de tempo (α_1, α_2).

$$g_n = \begin{cases} c_{1,} & n < \alpha_1, \\ \frac{c_2 - c_1}{\alpha_2 - \alpha_1}(n - \alpha_2) + c_2, & \alpha_1 \leq n < \alpha_2, \\ c_2, & n \geq \alpha_2. \end{cases} \tag{3.14}$$

Se $k > \alpha_2 - \alpha_1$, a diferença entre os quadros l e $l+k$ da transição g_n pode ser verificada pela equação 3.15:

$$D_{g_n}(l, l-k) = \begin{cases} 0, & n < \alpha_1 - k \\ \frac{|c_2 - c_1|}{|\alpha_2 - \alpha_1|}\left[n - (\alpha_1 - k)\right], & \alpha_1 - k \leq n < \alpha_2 - k \\ \mid c_2 - c_1 \mid, & \alpha_2 - k \leq n < \alpha_1 \\ -\frac{|c_2 - c_1|}{|\alpha_2 - \alpha_1|}(n - \alpha_2), & \alpha_1 \leq n < \alpha_2 \\ 0, & n \geq \alpha_2. \end{cases} \tag{3.15}$$

onde $D_{g_n}(l, l-k)$ corresponde a um planalto simétrico e o algoritmo de detecção de transição gradual visa identificar este padrão de planaltos.

A abordagem de Shen e Delp [38] utiliza a comparação de histogramas de cores utilizando termos DC de quadros consecutivos para a detecção de tomadas. Termos DC de quadros I são utilizados diretamente do *stream* MPEG e os termos DC de quadros P e B são reconstruídos por um algoritmo. As transições são detectadas através da geração de diagramas.

O diagrama da diferença do histograma é gerado utilizando-se a soma da diferença absoluta entre termos DC de imagens consecutivas, apresentada pela equação 3.6. Um corte é representado no diagrama por um simples pulso agudo e *dissolves* são representados por pulsos consecutivos médios e altos. Cortes são detectados com o uso de um limiar estático. Transições graduais são detectadas pela diferença do histograma do quadro corrente comparado à média da diferença de histogramas de quadros anteriores dentro de uma janela. Se a diferença é n vezes

maior que a média, isto indica um possível início de transição gradual. O mesmo valor de *n* é usado como um limiar suave para os quadros seguintes. O fim da transição é detectada quando a diferença do histograma é menor que o limiar [22].

Nesta abordagem a computação dos histogramas é mais rápida do que se fossem utilizados valores de *pixels* para tamanho original da imagem, porém ela não é capaz de distinguir uma transição gradual de movimentos rápidos de objetos. Um filtro da mediana pode ser aplicado para suavizar a diferença de histograma na detecção da transição gradual.

Existem outras variantes à abordagem baseada em termos DC. Uma delas utiliza a intersecção de histogramas de luminância e o cálculo do desvio padrão para a componente de luminância. Porém, técnicas baseadas em histogramas de luminância falham na detecção de transições, se a distribuição de luminosidade de quadros não muda significativamente.

Outra abordagem variante utiliza apenas termos DC de quadros I, computando histogramas de intensidade para termos DC e compara-os através da probabilidade de Yakimovski, do teste χ^2 e da estatística de Kolmogorov-Smirnov. Esta abordagem não necessita da reconstrução de termos DC, pois apenas quadros I são utilizados, entretanto, o exato limite da tomada não pode ser determinado.

3.2.3 Segmentação temporal de vídeo baseada em termos DC e modo de codificação de macrobloco

No algoritmo de detecção de limites de tomadas baseado em termos DC e um tipo de codificação MB [26], apenas componentes DC para quadros P são reconstruídas. A transição gradual é detectada pelo cálculo da variância σ^2 da seqüência de termos DC de quadros I e P e pela busca de parábolas nesta curva.

Uma transição gradual é uma mistura linear de duas seqüências de vídeo f_1 e f_2 com variância de intensidade σ_1 e σ_2, representada pela equação 3.16:

$$f(t) = f_1(t)\,[1 - \alpha(t)] + f_2(t)\alpha(t) \tag{3.16}$$

onde $\alpha(t)$ é um parâmetro linear e a forma parabólica da curva de variância é representada pela equação abaixo:

$$\sigma^2(t) = (\sigma_1^2 + \sigma_2^2)\alpha(t) - 2\sigma_1^2\alpha(t)$$

Cortes são detectados pelo cálculo das três taxas abaixo:

$$R_p = \frac{intra}{forw},\ R_b = \frac{back}{forw},\ R_f = \frac{forw}{back}$$

onde *intra*, *forw* e *back* são o número de macroblocos no quadro corrente que possuem, respectivamente, codificação interna, posterior e anterior.

Quando existe um corte em um quadro P, conseqüentemente, não existem muitos MBs de quadros anteriores para compensação de movimento e muitos MBs são intra-codificados. Assim, um corte em um quadro P é detectado se existe pico em R_p e se existe um corte em um quadro B, a codificação será relativa ao quadro anterior. Entretanto, um corte em um quadro B é detectado se existe um pico em R_b. Um quadro I é um suspeito de corte se existe um pico em $|\Delta\sigma^2|$ para este quadro, pois a variância de intensidade do quadro durante uma tomada é estável, e se quadros B anteriores a I tiverem picos em R_f [22].

3.2.4 Segmentação temporal de vídeo baseada em coeficientes DCT, modo de codificação MB e MVs

Zhang et al. [47] propõem a localização das regiões de possíveis transições e movimentos de câmera e objetos, aplicando-se a equação da diferença de coeficientes DCT de quadros I, como exibido na equação 3.12.

Em seguida, deve-se confirmar os cortes detectados anteriormente e detectar a sua exata localização, checando o número M de vetores de movimento (MV) para áreas selecionadas. Sendo M o número de MVs em quadros P e o menor dos números de MVs não zerados, com codificação anterior e posterior de quadros B. Um corte é detectado antes ou depois do quadro B e P, se $M < T$ (T é um limiar perto de zero). Transições graduais são encontradas pela adaptação do algoritmo *twin comparison*, utilizando-se a diferença de DCT de quadros I.

Este técnica utiliza apenas informações disponíveis diretamente no *stream* MPEG. Proporciona, também, alta velocidade de processamento, boa precisão e detecta falsos positivos em caso de quadros estáticos, mas não diferencia transições graduais de movimentos de objetos.

3.2.5 Segmentação temporal de vídeo baseada em modo de codificação de macrobloco e vetores de movimento

A abordagem de Koprinska e Carrato [21] é baseada em um conjunto de regras e um módulo de rede neural. Uma busca superficial procura picos nos macroblocos intra-codificados de quadros P. Picos agudos indicam cortes e picos graduais com uma forma específica indicam transições graduais. Em seguida, uma busca precisa nos quadros da vizinhança, refina a solução.

A busca precisa revela cortes que permaneceram escondidos na busca superficial. As regras para a localização de cortes são baseadas no número de MBs com codificação anterior e posterior. Para a detecção de *fades* de borda preta, utiliza-se o número de MBs interpolados e com codificação anterior. A rede neural é usada para diferenciar movimentos de objeto e câmera e encontrar a localização dos limites de uma transição gradual. A rede aprende com exemplos pré-classificados na forma do padrão de MV, correspondendo as classes: estacionária, panorama, *zoom*, movimento de objeto, *tracking* e *dissolve*s [22].

Esta abordagem é rápida e robusta para operações de câmera, e precisa na localização de cortes, *fades* e *dissolves* simples. Entretanto, alguns *dissolves* entre seqüências movimentadas são reconhecidas como movimento de objetos e seus limites não são determinados.

3.2.6 Segmentação temporal de vídeo baseada em modo de codificação MB e informação de taxa de bit

Feng et al. [15] sugerem uma técnica para a detecção de cortes que utiliza a informação de taxa de bit no nível de MB e o número de movimentos previstos em MBs. Um corte é detectado se existe uma grande mudança na taxa de bit entre dois quadros I ou P consecutivos. O número de MBs com codificação anterior é usado para detectar cortes em quadros B. Então, a taxa é calculada pela equação.

$$R_b = \frac{back}{mc} \tag{3.17}$$

sendo *back* o número de MBs com codificação anterior e *mc* o número de MBs com todos os movimentos compensado em um quadro B.

3.3 Tomografia de vídeo

Algumas técnicas analisam seqüências espaço-temporais de imagens de vídeo para a identificação de alguns de seus eventos. A abordagem de Akutsu e Tonomura [1] sugere a criação de duas imagens, chamadas raio-x (*x-ray*) e raio-y (*y-ray*), a partir de seqüências espaço-temporais de imagens de vídeo, com o propósito de fornecer padrões de identificação de operações de câmera.

A imagem raio-x é obtida fixando-se o eixo y como constante durante uma seqüência de vídeo e a imagem raio-y é obtida fixando-se x como constante. Então, são aplicados um filtro de arestas e a transformada de *Hough* sobre as imagens para a obtenção de algumas operações de câmera.

3.4 Ritmo visual por amostragem

Recentemente, novas abordagens para a segmentação de vídeo utilizam a análise de uma única imagem que representa todo o segmento de vídeo. Esta imagem de representação do vídeo é chamada de ritmo visual [20] [11] [16] ou espaço-temporal [29].

Estas técnicas diferenciam-se das demais por não utilizarem medidas baseadas na dissimilaridade de quadros e, ao invés disso, buscam padrões em uma imagem criada a partir de uma amostra de cada quadro de uma seqüência de vídeo, preservando muitas das características do vídeo original [16].

A amostragem de cada quadro é obtida através da extração de uma fatia diagonal, vertical ou horizontal de cada quadro, como ilustrado na figura 3.2. Uma imagem é criada utilizando-se estas amostras e, assim, o conteúdo do vídeo sofre uma simplificação, pois cada fatia é transformada em uma vertical da imagem.

Esta imagem construída é chamada ritmo visual por amostragem (figura 3.3 na página seguinte) e é capaz de representar todo o conteúdo do vídeo. A largura do ritmo visual corresponde à mesma quantidade de quadros do segmento de vídeo.

O ritmo visual inclui características visuais que permitem a distinção e classificação de diferentes tipos de efeitos de vídeo: cortes, *wipes*, *dissolves*, *fades*, movimentação de câmera e objetos, *flashes* e *zooms* [20].

Cada um destes efeitos produzem diferentes padrões no ritmo visual e, portanto, para se detectar edições de vídeo, métodos de processamento de imagens são aplicados, visando a

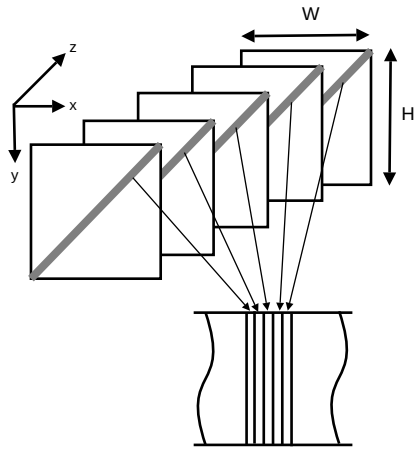

Figura 3.2: Exemplo de ritmo visual usando a diagonal de cada quadro

Figura 3.3: Exemplo de uma imagem de ritmo visual obtida pela amostragem da diagonal principal

identificação das diferentes classes de padrões existentes no ritmo visual.

Definição 3.1. Ritmo Visual Segundo a definição de Guimarães [16], seja $V = (f_t)_{t\in[0,Tempo-1]}$ um segmento de vídeo, no domínio $2D + t$. O ritmo visual, no domínio $1D + t$, é uma simplificação do vídeo em que cada quadro f_t é transformado em uma linha vertical da imagem de ritmo visual A, definida como na equação 3.18:

$$VR(t, z) = f_t(r_x \times z + a, r_y \times z + b) \tag{3.18}$$

onde $z \in \{0, ..., H_A - 1\}$ e $t \in \{0, ..., Tempo - 1\}$, H_A e $Tempo$ são a altura e a largura do ritmo visual, respectivamente, r_x e r_y são as razões da amostragem de *pixel*, a e b são deslocamentos em cada quadro.

Pela definição de ritmo visual acima, diferentes amostragens de *pixels* dos quadros poderiam ser utilizadas. Seja H a altura e W a largura de cada quadro, poderiam-se obter todos os *pixels* da diagonal principal dos quadros, se $r_x = r_y = 1$, $a = b = 0$ e $H = W$. Todos os *pixels* da diagonal secundária poderiam ser obtidos, se $r_x = -1$, $r_y = 1$, $a = H$, $b = 0$ e $H = W$. Uma linha central horizontal poderia ser obtida se $r_x = 1$, $r_y = 0$, $a = 0$ e $b = W/2$. Os *pixels* da linha vertical central poderiam ser obtidos, se $r_x = 0$, $r_y = 1$, $a = H/2$ e $b = 0$. A figura 3.4 exemplifica as diferentes amostragens de *pixels*.

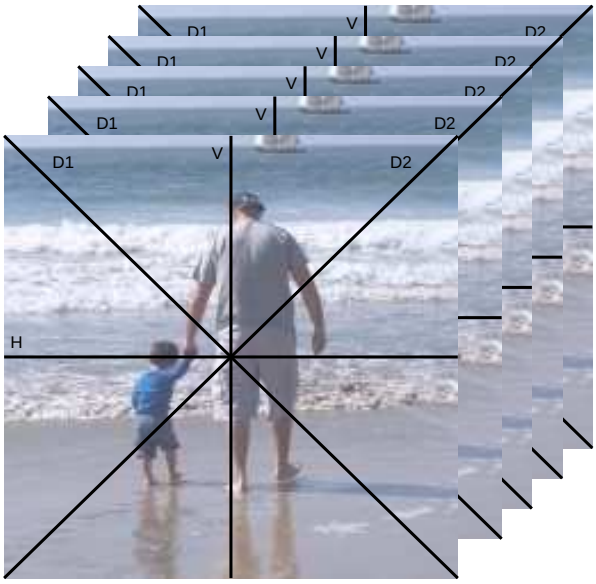

Figura 3.4: Exemplo de tipos de amostragens de *pixels*

Com a simplificação do segmento de vídeo digital em uma imagem de ritmo visual, é possível visualizar os quadros onde ocorrem as transições através da identificação de padrões

específicos. Diferentes amostragens produzem diferentes ritmos visuais e, conseqüentemente, os eventos de vídeo são identificados por diferentes padrões. Porém, a diagonal principal dos quadros fornece as melhores características visuais para distinguir as transições, pois esta possui características horizontais e verticais da imagem [20].

Utilizando-se a diagonal principal, cortes podem ser identificados pela detecção de linhas verticais divisórias na imagem de ritmo visual, como pode ser verificado na figura 3.5 (a). O padrão de *wipes* (figura 3.5 (b)) é semelhante ao padrão de cortes, e podem ser detectados através de uma linha divisória inclinada na imagem de ritmo visual [16].

Ainda utilizando-se a diagonal principal como amostragem, um *dissolve* pode ser identificado por um limite borrado no ritmo visual [29], pois é resultado de uma transição lenta entre duas regiões não monocromáticas (figura 3.5 (c)). Regiões verticais claras e estreitas no ritmo visual podem representar *flashes*, como exemplificado na figura 3.6.

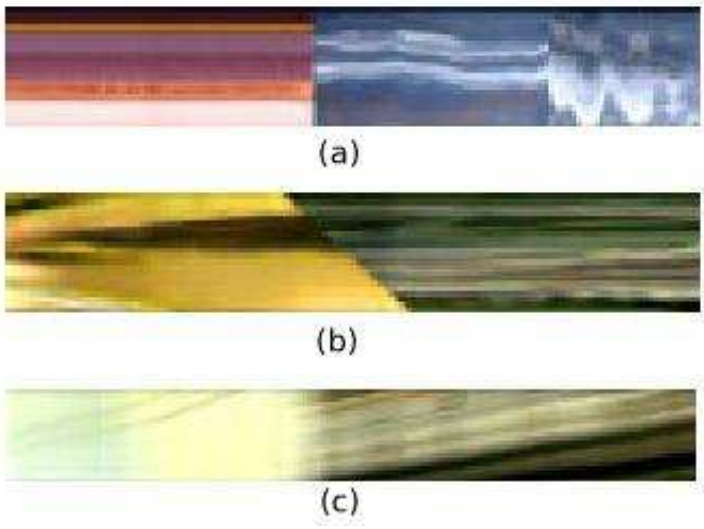

Figura 3.5: Exemplos de transições presentes no ritmo visual: (a) Três tomadas de câmera conectadas por dois cortes; (b) Duas tomadas conectadas por um *wipe*; (c) Duas tomadas conectadas por um *dissolve* [29].

Segundo Guimarães, o padrão que representa um *fade* é identificado por uma transição gradual entre uma região monocromática e uma região não monocromática, como exemplificado na imagem 3.7. É possível, ainda, detectar-se operações de câmera como *zoom-in*, *zoom-out* e *pan* através da identificação de regiões expandida, afunilada e deslocada, respectivamente, no ritmo visual por amostragem (figura 3.8 na página seguinte).

A detecção dos eventos de vídeo requer a identificação de seus padrões no ritmo visual. No método proposto por Guimarães, a detecção de alguns padrões é realizada a partir da

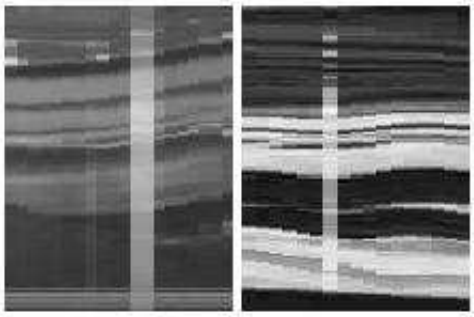

Figura 3.6: Exemplo de *flashes* [16].

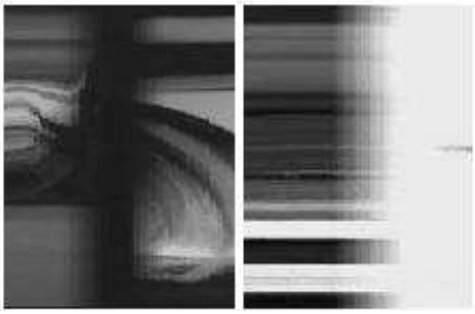

Figura 3.7: Exemplo de *fades* [16].

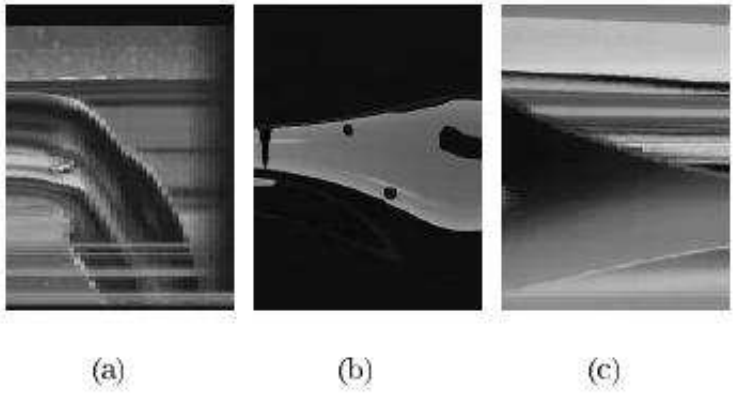

Figura 3.8: Exemplo de regiões deformadas presentes no ritmo visual: (a) *pan*; (b) *zoom-in*; (c) *zoom-out* [16].

morfologia matemática em níveis de cinza, topologia digital e geometria discreta.

3.5 Ritmo visual por histograma

O ritmo visual por histograma proposto por Guimarães tenta utilizar-se das vantagens existentes na utilização de histogramas, tais como informação global, invariância à rotação e translação da imagem. Nesta abordagem, ao invés de se obter uma amostra de cada quadro, obtém-se o histograma de cada quadro para formar a imagem de ritmo visual.

Definição 3.2. Ritmo Visual por Histograma – Segundo a definição de Guimarães, seja $V = (f_t)_{t\in[0,Tempo-1]}$ um segmento de vídeo, no domínio $2D + t$ e $(H_{ft})_{t\in[0,Tempo-1]}$ os histogramas de cada quadro de V. O ritmo visual por histograma B (3.19) é uma imagem 2D onde cada linha vertical representa um histograma de um quadro:

$$B(t, z) = H_{ft}(z) \tag{3.19}$$

onde $t \in \{0, ..., Tempo - 1\}$ e $z \in \{0, L - 1\}$, $Tempo$ é o número de quadros e L o número de pacotes do histograma.

A maior dificuldade desta abordagem está relacionada à conversão de todos os valores do histograma em níveis de cinza. Assim sendo, cada histograma é normalizado independentemente para a representação do ritmo visual, causando um efeito de filtragem dos menores valores do histograma.

A identificação de transições ocorre de maneira análoga ao ritmo visual por amostragem. Cortes são representados por linhas verticais e linhas inclinadas podem representar *fades*, que também podem ser representados por regiões deformadas. *Flashes* são representados por uma descontinuidade ortogonal presente no ritmo visual por histograma.

Diferentemente do ritmo visual por amostragem, as regiões deformadas não representam operações de câmera, mas são associadas às transições graduais. Regiões expandidas e afuniladas representam *fades* e regiões *fuzzy* representam *dissolves*. O método para a identificação automática dos padrões é semelhante ao aplicado no ritmo visual por amostragem, envolvendo morfologia matemática, topologia digital e geometria discreta.

A principal desvantagem desta abordagem é que o tempo computacional para a obtenção do ritmo visual por histograma é maior que o tempo para obtenção do ritmo visual por amostragem.

Capítulo 4

Metodologia

4.1 Introdução

O presente trabalho trata da análise de transições abruptas de vídeo, baseada na abordagem de ritmo visual por amostragem de Guimarães [16], descrita na seção 3.4 desta dissertação. Esta abordagem realiza a análise de fatias espaço-temporais de cada quadro, que são extraídas para se obter uma simplificação do vídeo na imagem de ritmo visual (figura 4.1).

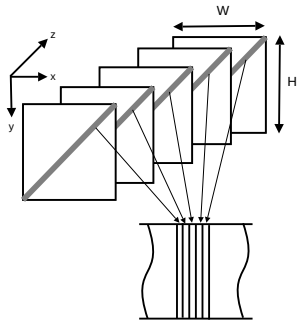

Figura 4.1: Ritmo visual obtido utilizando-se a diagonal de cada quadro

A abordagem de ritmo visual proposta por Guimarães necessita da descompressão prévia do vídeo a ser analisado e da sua conversão para níveis de cinza. Neste aspecto, esta abordagem apresenta a limitação de uso em sistemas em tempo real, que é uma característica desejada em muitas aplicações multimídia.

Como a atual disponibilidade de vídeos comprimidos é enorme, torna-se inadequado o emprego de sua descompressão prévia. Nesta pesquisa são utilizados vídeo comprimidos, trabalhando-se diretamente no domínio de compressão MPEG. A escolha do formato de compressão MPEG se deve ao fato de ser o padrão mais aceito internacionalmente para a com-

pressão de vídeo digital.

A imagem 4.2 na página seguinte apresenta uma visão geral da metodologia proposta por esta dissertação.

4.2 Ambiente de desenvolvimento

O sistema foi desenvolvido na plataforma Linux, devido à boa disponibilidade do ambiente e bibliotecas para a programação. Foram utilizados o compilador *GNU Project C and C++ Compiler* (GCC) e o depurador *GNU Debugge*r (GDB).

A implementação foi realizada em linguagem C++, utilizando-se o *Open Source Computer Vision* (OpenCV), composto por um conjunto de bibliotecas de manipulação de imagens que auxiliam o desenvolvimento de aplicações de visão computacional. Esta biblioteca facilitou a manipulação das imagens no modelo de cores HSV e a implementação da morfologia em cores.

Para auxiliar na manipulação das estruturas do vídeo MPEG, bem como a criação das imagens DC e identificação de quadros I, foi utilizada o *ffmpeg*, que é uma biblioteca de manipulação de áudio e vídeo.

4.3 Ritmo visual

A metodologia proposta, inicia-se pela construção da imagem de ritmo visual. O ritmo visual é formado a partir da extração dos *pixels* da diagonal principal de miniaturas de cada quadro (imagens DC). As imagens DC são construídas pelo processamento dos termos DC dos quadros, como descrito na seção 2.4 desta dissertação. Para a manipulação das estruturas do MPEG e formação das imagens DC, foi utilizada a biblioteca *ffmpeg*, facilitando muito toda a implementação relativa ao MPEG.

4.3.1 Largura da fatia

Neste trabalho, é realizado um estudo sobre a influência da largura da amostra de cada quadro na precisão da detecção de transições. Este estudo revela que uma largura de fatia maior não aumenta a precisão da detecção. Para tanto, a solução desenvolvida foi testada para ritmos visuais com fatias de um e três *pixels* de largura, como pode ser verificado na imagens 4.2 e 4.3.

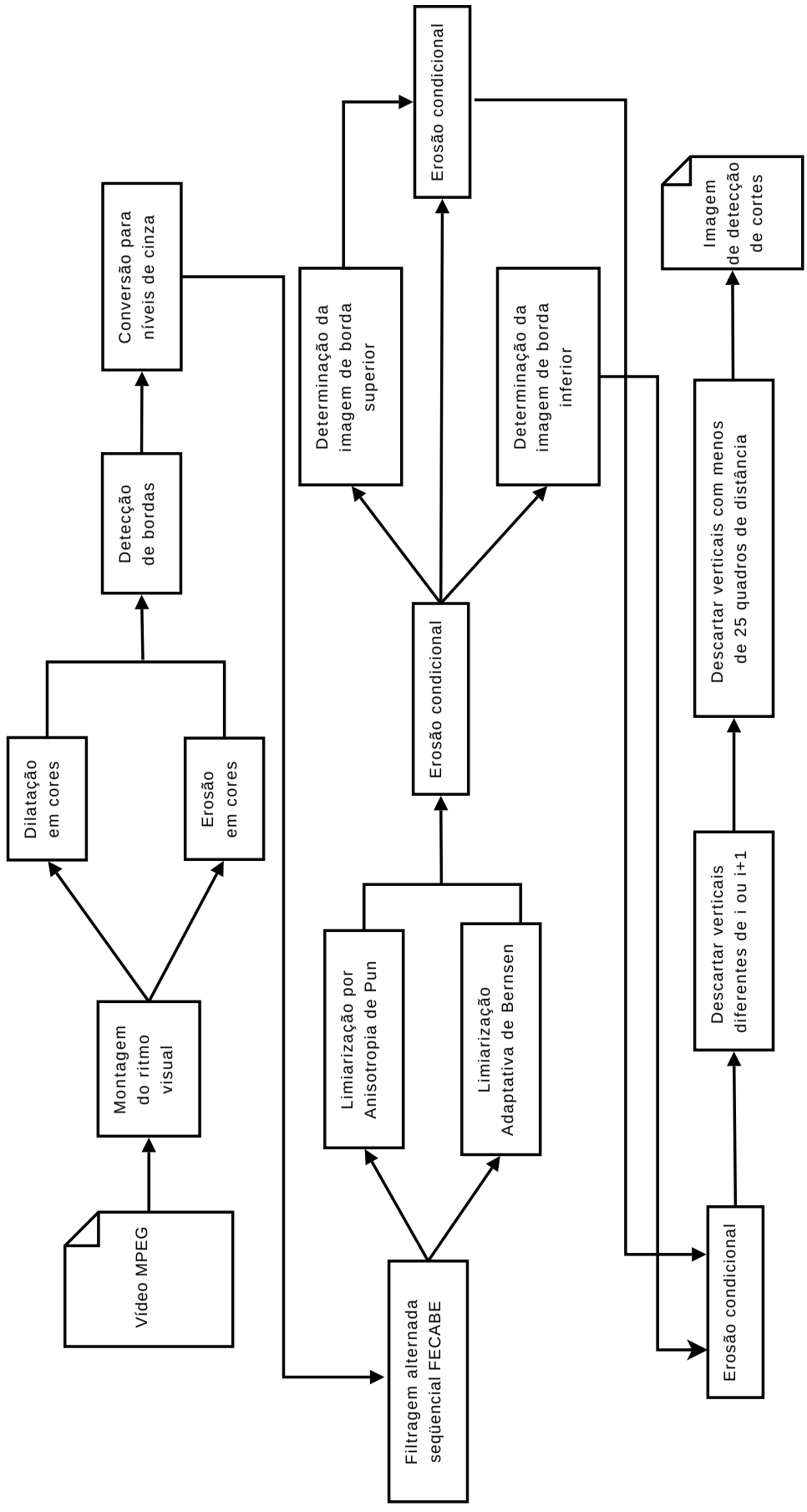

Figura 4.2: Visão geral da metodologia

Figura 4.3: Exemplo de uma imagem de ritmo visual obtida pela amostragem da diagonal principal com 1 *pixel* de largura

Figura 4.4: Exemplo da mesma imagem de ritmo visual anterior obtida pela amostragem da diagonal principal com 3 *pixels* de largura

A melhor detecção de cortes no ritmo visual ocorre quando este é montado a partir de fatias de um *pixel* de largura, pois esta é uma amostra suficiente para se detectar o corte. E, por conseqüência, quando mais *pixels* desnecessários são inseridos, falsos positivos podem ocorrer. Em testes com largura de fatias de três *pixels*, alguns falsos positivos são inseridos pela adição de *pixels* desnecessários, apenas dificultando a identificação dos cortes no ritmo.

No exemplo da figura 4.6 na próxima página, dois trechos de ritmo visual foram criados a partir de três *pixels* de largura de quadro. Nesta imagem, pode-se verificar que alguns cortes falsos foram inseridos (a parte do sistema relativo à redução de falsos positivos não foi aplicada - seção 4.8). Contudo, para o ritmo formado por um *pixel* de largura, como exemplificado pela figura 4.5, estas falsas linhas não foram inseridas.

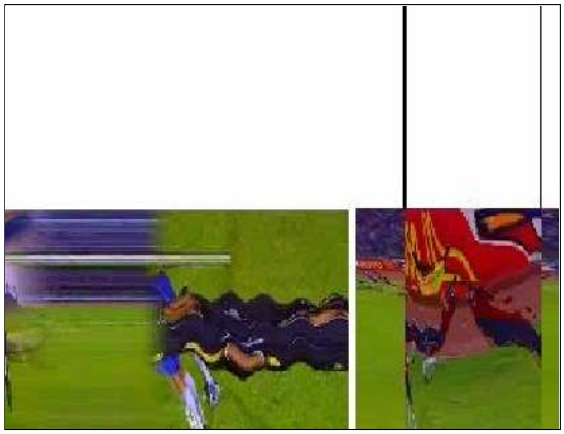

Figura 4.5: Detecção de cortes aplicada em dois trechos de ritmo visual formados a partir de 1 *pixel* de largura de quadro.

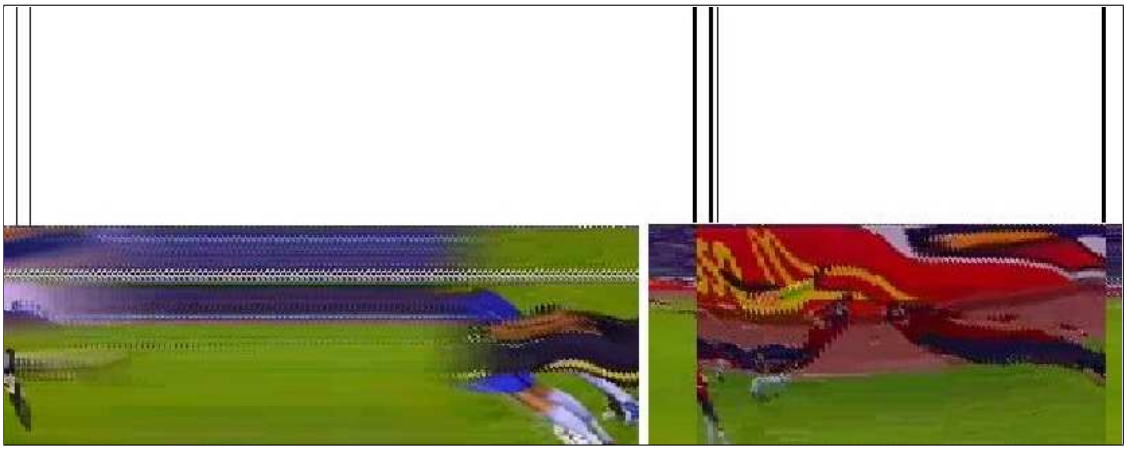

Figura 4.6: Detecção de cortes aplicada a dois trechos de ritmo visual formados a partir de três *pixels* de largura de quadro.

4.3.2 Amostra vertical, horizontal ou diagonal

Segundo Kim et al. [20], a diagonal principal fornece as melhores características visuais para a detecção de transições no ritmo visual. Apesar disto, faz-se necessário um estudo da melhor opção para o caso da base de vídeos escolhida (vídeos de jogos de futebol). Desta forma, foram realizados testes utilizando a linha horizontal central, a vertical central e a diagonal principal de cada quadro para a formação do ritmo visual.

Estes testes confirmam que, para a detecção de cortes em seqüências de jogos de futebol, a melhor opção para a construção do ritmo visual é a diagonal principal. Esta opção se deve ao fato de que linhas verticais e horizontais, muitas vezes, são paralelas às retas de marcação do campo, e em alguns pontos, são destacadas como se fossem cortes. Já a diagonal principal, dificilmente criaria um falso corte advindo das linhas de marcação de campo, pois não é paralela a nenhuma delas.

Ritmos visuais baseados na linha horizontal central de cada quadro destacam, erroneamente, como corte, as linhas horizontais de demarcação do campo. Como conseqüência, a opção de criação do ritmo visual utilizando linhas horizontais foi descartada para jogos de futebol. Na imagem 4.7 na próxima página é possível verificar uma linha vertical branca do campo, que visualmente aparenta ser um corte.

Da mesma forma, o ritmo visual criado a partir da linha vertical de cada quadro destaca as marcações verticais existentes no campo, criando linhas no ritmo visual que levarão a falsos positivos, como pode ser verificado na imagem 4.8 na página seguinte. Desta forma, esta opção também foi descartada para jogos de futebol.

Figura 4.7: Ritmo visual obtido utilizando-se a linha horizontal central

Figura 4.8: Ritmo visual obtido utilizando-se a linha vertical central

Portanto, a imagem de ritmo visual da metodologia proposta, é construída extraindo-se a diagonal principal (com um *pixel* de largura) de cada imagem DC, formando uma linha vertical na imagem de ritmo visual.

4.3.3 Montagem do ritmo visual

Como cada imagem DC é uma versão reduzida do quadro, a utilização desta miniatura, ao invés do quadro original, resulta em maior rapidez na criação do ritmo visual e no seu processamento em busca de cortes.

Neste trabalho, utiliza-se uma redução de 50% de cada quadro para a construção do ritmo visual, obtendo-se uma rapidez significativa de processamento, visto que, desta forma, processa-se uma quantidade menor de todo o vídeo. Esta redução pode ser facilmente obtida através da biblioteca *ffmpeg*. Optamos pela utilização de uma redução de 50% ao invés de 1/8 do quadro original para facilitar na análise dos resultados. Contudo, a solução ideal seria a utilização de miniaturas de 1/8 do tamanho original, visando maior redução no tempo de processamento.

Nas imagens 4.9 na próxima página e 4.10 na página seguinte são demonstrados um ritmo visual criado a partir de quadros com 50% do seu tamanho e um ritmo visual construído

utilizando-se quadros do tamanho original.

Figura 4.9: Ritmo visual obtido utilizando quadros em tamanho reduzido

Figura 4.10: Ritmo visual obtido utilizando quadros em tamanho normal

Algumas abordagens de segmentação de vídeo baseadas em termos DC utilizam apenas os quadros I (*intra frame*, seção 2.4) de vídeo MPEG, reduzindo-se consideravelmente o tempo de processamento do algoritmo. Os quadros I são codificados sem nenhuma predição temporal e ocorrem quando existem grandes mudanças de cena. Desta forma, o uso de quadros I para auxiliar na identificação de transições abruptas baseia-se no fato de haver uma mudança significativa entre quadros na existência de um corte.

Neste trabalho, a imagem de ritmo visual é criada a partir de todos os quadros para facilitar a identificação e localização de cada transição na fase de testes. Mas, a informação da localização dos quadros I é utilizada no algoritmo de detecção de cortes, visando diminuir o número de falsos positivos, descartando-se os outros quadros.

Acredita-se que os mesmos resultados podem ser obtidos criando-se o ritmo visual utilizando, apenas, os quadros I, pois estes mostram-se eficientes na detecção de cortes. A imagem 4.12 na próxima página foi construída utilizando-se todos os quadros do vídeo. Já a

imagem 4.11 representa o mesmo segmento de vídeo, porém construída utilizando-se apenas quadros I na formação do ritmo visual. Comparando-se as duas imagens, é possível identificar que os quatro cortes existentes na imagem 4.12 são mantidos na imagem 4.11.

Figura 4.11: Ritmo visual obtido utilizando-se apenas quadros I

Figura 4.12: Ritmo visual obtido utilizando-se todos os quadros

4.4 Morfologia em cor empregada

Depois da criação do ritmo visual, técnicas de morfologia matemática são empregadas para a identificação automática dos padrões de detecção de cortes. Como são utilizadas a diagonal principal para a formação do ritmo visual, os cortes são detectados a partir da identificação de padrões verticais presentes nos ritmos visuais, como visto na figura 3.5 na página 51.

No método proposto por Guimarães, a segmentação do ritmo visual é realizada empregando-se a morfologia matemática em níveis de cinza, utilizando-se vídeos convertidos para níveis de cinza. Nesta dissertação, uma nova abordagem é proposta, na qual o ritmo visual é processado em cores, empregando-se a morfologia em cores para identificação dos padrões.

Esta técnica oferece uma redução das perdas ocorridas na conversão do vídeo para níveis de cinza. A figura 4.13 na próxima página exemplifica um ritmo visual em cores obtido pela extração da diagonal principal de cada quadro. Apesar de representar uma redução de complexidade, a conversão para níveis de cinza pode, também, distorcer ou confundir o conteúdo das informações, o que não ocorreria utilizando-se a imagem colorida. Estas perdas podem

ser visivelmente verificadas no exemplo da figura 4.14, que representa o mesmo ritmo visual convertido para níveis de cinza.

Figura 4.13: Ritmo visual colorido.

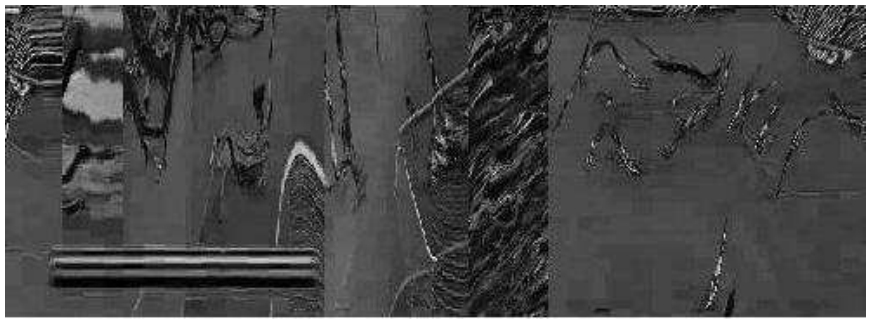

Figura 4.14: Ritmo visual em níveis de cinza.

Segundo Calixto [9], é possível a construção de uma relação de ordem em um determinado espaço de cor e, assim, definir uma morfologia em cores. Para utilizar esta morfologia, a ordenação de cores emprega a constante de cromaticidade, como visto na seção 2.6. Esta técnica exige a definição de uma cor, geralmente do fundo, como sendo a mínima, para melhor realizar a ordenação das cores.

Para não haver a inversão de operadores morfológicos, a cor de fundo dos quadros é escolhida como cor mínima. Como a cor do fundo é, normalmente, a cor predominante no quadro, determinamos a cor mínima indentificando a cor predominante no histograma de cores do ritmo visual.

Sobre a imagem de ritmo visual colorida são aplicadas as técnicas de dilatação e erosão coloridas (seção 2.6), com elemento estruturante vertical e uma iteração. A escolha do elemento estruturante vertical deve-se ao fato desta estrutura ressaltar as linhas verticais, que é o padrão para a identificação de cortes. Então, a detecção de bordas é aplicada, subtraindo-se a imagem dilatada da imagem erodida. Assim, uma imagem semelhante a figura 4.15 na próxima página é obtida.

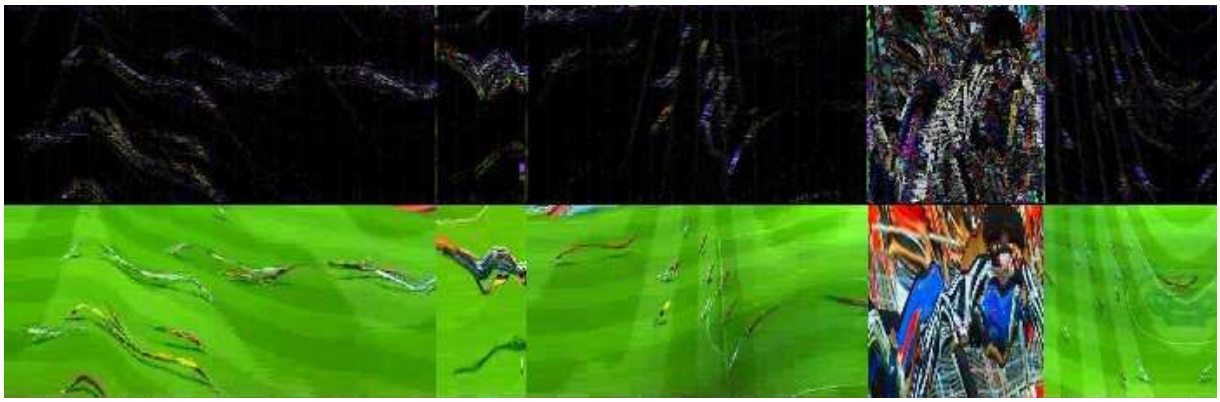

Figura 4.15: Ritmo visual e a imagem de detecção de bordas correspondente.

Após a detecção de bordas, a valiosa informação presente na imagem colorida já teve sua utilidade, destacando os padrões desejados. Neste ponto de processamento, a quantidade de informação é muito menor e, assim sendo, a imagem de detecção de bordas é convertida para níveis de cinza.

4.5 Filtragem

Nesta fase de processamento, uma filtragem progressiva aplicada à imagem de detecção de bordas do ritmo visual faz-se necessária, objetivando a eliminação de ruídos na imagem. Esta filtragem progressiva parte, inicialmente, de um elemento estruturante menor e vai crescendo conforme o número de iterações. A aplicação do filtro alternado seqüencial FECABE (subseção 2.5.9) visa destacar as regiões mais claras da imagem, eliminando o ruído escuro e integrando o ruído claro às regiões claras [14].

Portanto, o filtro alternado seqüencial FECABE com onze iterações e elemento estruturante vertical é aplicado à imagem de detecção de bordas, tendo em vista destacar os padrões verticais, como pode ser verificado na imagem 4.16.

Figura 4.16: Filtro seqüencial FECABE aplicado à imagem de detecção de bordas.

A quantidade de iterações utilizadas na etapa da filtragem alternada seqüencial FECABE

foi definida de forma empírica, ou seja, testes com diferentes número de iterações foram realizados e verificou-se que os melhores resultados de precisão e revocação (seção 5.5) são obtidos utilizando-se onze iterações, conforme pode ser verificado no exemplo da tabela 4.1.

Tabela 4.1: Testes de variação do número de iterações na filtragem alternada seqüencial FECABE - vídeo Atlético x Botafogo - 1^o tempo

Iterações x Resultados		
Iterações	**Precisão**	**Revocação**
5	64%	85%
8	68%	86%
10	70%	87%
11	73%	89%
12	73%	85%

4.6 Limiarização

Após a filtragem, duas limiarizações são empregadas sobre a imagem filtrada. A primeira, Limiarização Global por Anisotropia de Pun (subseção 2.7.2), visa obter uma maior quantidade de informações para que esta imagem limiarizada sirva como máscara na etapa de reconstrução. A imagem 4.17 é resultado da Limiarização por Anisotropia aplicada à imagem filtrada.

Figura 4.17: Limiarização de Anisotropia aplicada à imagem filtrada.

A segunda limiarização, Local Adaptativa de Bernsen (subseção 2.7.3), é empregada com contraste de 35 e janela de 30, objetivando uma imagem de marcador que limite o espaço de reconstrução da imagem. A imagem 4.18 na página seguinte é resultado da Limiarização Local de Bernsen aplicada à imagem filtrada.

Figura 4.18: Limiarização de Bernsen aplicada à imagem filtrada.

4.7 Erosão condicional

Após as limiarizações, as imagens são invertidas e a erosão condicional (subseção 2.5.4) é executada, utilizando-se a imagem limiarizada por Anisotropia de Pun como marcador e a limiarizada de Bernsen como máscara. Nesta etapa, como pode ser observado na figura 4.19, o resultado destaca apenas as linhas verticais.

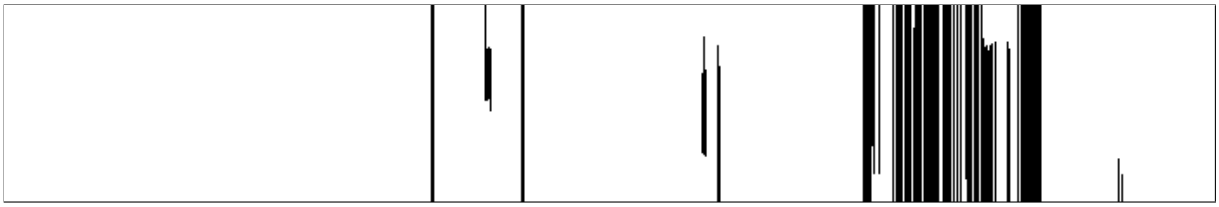

Figura 4.19: Resultado da erosão condicional utilizando a imagem limiarizada de Anisotropia como marcador e Bernsen como máscara.

Porém, o resultado obtido possui algumas linhas que não são contínuas e não tocam a borda superior e inferior da imagem. Como conseqüência, é necessário aplicar uma outra erosão condicional para que apenas as linhas que tocam a borda superior e inferior, sejam reconstruídas.

Assim sendo, primeiramente, a erosão condicional é executada com elemento estruturante vertical, utilizando-se a imagem de borda inferior como marcador e a imagem erodida, obtida anteriormente, como máscara. Na imagem 4.20, pode-se perceber que as linhas que não tocam a borda inferior são eliminadas.

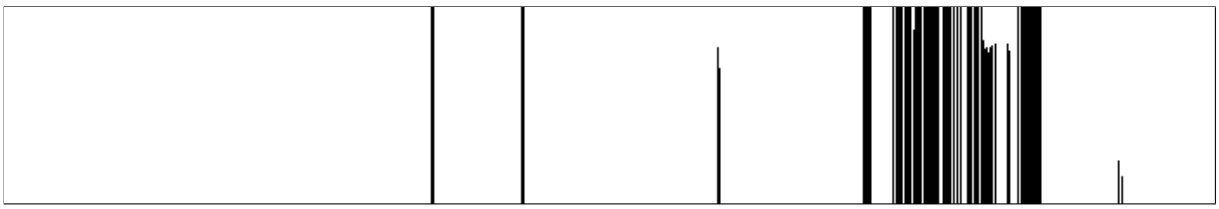

Figura 4.20: Resultado da erosão condicional utilizando a imagem de borda inferior como marcador e a imagem erosão como máscara.

Neste momento, faz-se necessário o emprego da erosão condicional com elemento estru-

turante vertical, utilizando-se a imagem de borda superior como marcador e a imagem erodida anterior como máscara. Nesta etapa, as linhas que não tocam a borda superior são eliminadas, como exemplificado na imagem 4.21.

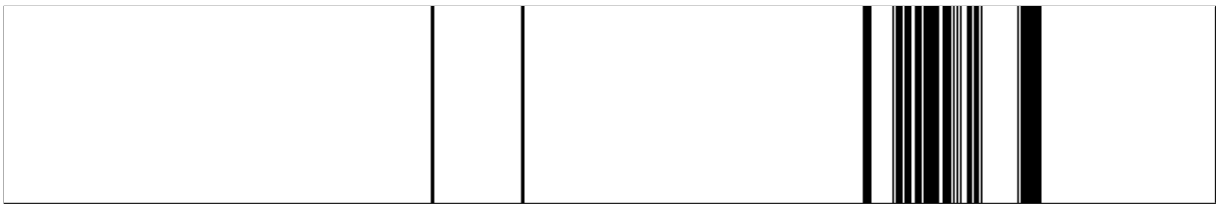

Figura 4.21: Resultado da erosão condicional utilizando a imagem de borda superior como marcador e a imagem erodida anteriormente como máscara.

A erosão condicional, utilizando imagens de borda como marcador, possui papel fundamental para que sejam obtidas apenas as linhas contínuas em contato com a borda superior e inferior, sem a necessidade da determinação de limiares para a detecção das linhas de corte.

4.8 Redução de falsos positivos

Nesta fase de processamento é possível obter as linhas indicadoras da localização dos cortes. Contudo, a base de testes escolhida oferece algumas dificuldades adicionais para a detecção destas transições. Os jogos de futebol geram muitos falsos positivos por alguns aspectos, como melhor visto no capítulo 6. Muitas operações de *zoom* são utilizadas para obtenção de particularidades dos lances ou da torcida. Estas aproximações geram linhas verticais que não correspondem a cortes reais, como pode ser verificado na imagem 4.22.

Figura 4.22: Resultado da erosão condicional utilizando a imagem de borda superior como marcador e a imagem erodida como máscara.

Outro fator agravante, é a grande quantidade de cortes de difícil detecção, que exigem

um processamento mais abrangente visando a sua identificação. Contudo, este processamento para a busca de cortes difíceis, é, também, responsável pela inserção de falsos positivos indesejados.

Desta forma, é necessária uma segunda etapa de processamento, responsável pela diminuição dos falsos positivos. É possível melhorar os resultados utilizando-se a informação da localização dos quadros I. Assim, as linhas verticais que não correspondem a um quadro I ou quadro I+1 são descartadas, como pode ser observado na imagem 4.23.

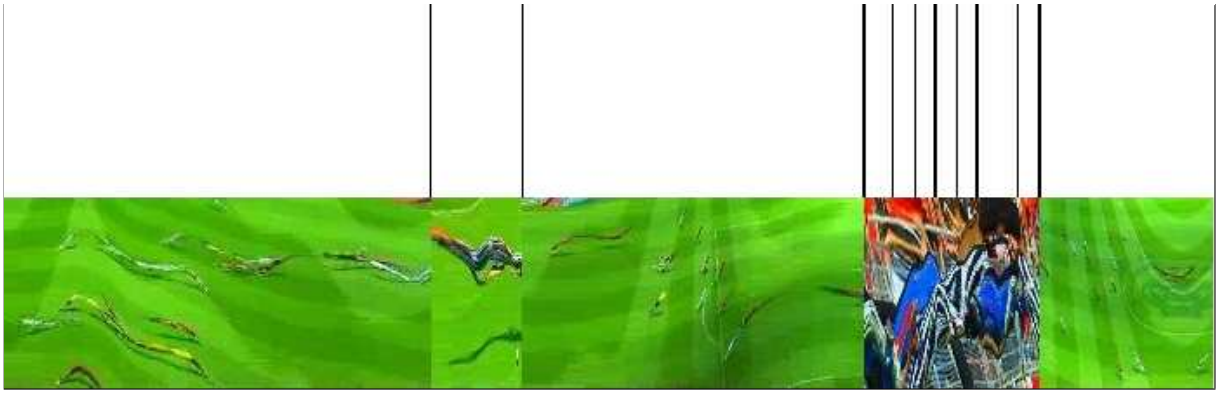

Figura 4.23: O algoritmo descarta as linhas que não correspondem a quadros I ou I+1.

Se um vídeo de jogo de futebol possui em média 30 quadros por segundo, seria difícil a existência de dois cortes em menos de um segundo, pois estes cortes não fariam sentido no contexto do jogo. Neste aspecto, outra melhoria é inserida ao algoritmo visando, também, a diminuição de falsos positivos.

Os cortes com menos de 25 quadros de distância (valor inferior a um segundo de distância entre cortes) são descartados e mantém-se apenas a primeira e última linha nos casos de identificação de *zoom* (várias verticais próximas). Um exemplo de operação de *zoom* pode ser observado na imagem 4.23, onde há várias linhas verticais bem próximas. Se o tipo de programação analisada utiliza as operações de *zoom*, esta condição inserida ao algoritmo resulta em uma melhoria significativa, pois diminui-se muito o número de falsos positivos, como pode ser verificado na figura 4.24.

Porém, existem programações como *trailer* de filme ou certos comerciais de televisão, em que vários cortes em pequenos espaços de tempo são inseridos, visando uma sensação de velocidade à cena. A imagem de ritmo visual 4.25 foi obtida a partir de um vídeo de comercial, em que cortes com menos de um segundo são intencionais. Por esta razão, esta segunda etapa do algoritmo não deve ser aplicada a vídeos que apresentam esta característica.

Figura 4.24: O algoritmo descarta os cortes com distância maior que 25 quadros.

Figura 4.25: Ritmo visual de um comercial de televisão com cortes com menos de um segundo de distância.

Capítulo 5

Experimentos e Resultados

5.1 Introdução

Após o desenvolvimento do sistema, foram necessárias as fases de treinamento, validação e testes. Em uma primeira etapa, realizou-se a seleção de vídeos digitais para compor uma base de vídeos. Esta base supre os conjuntos de treinamento, validação e testes.

A fase de treinamento tem como finalidade o ajuste do sistema, para que este atinja os resultados esperados. A fase de validação é empregada para validar e verificar a eficiência do método proposto, corrigindo-se o sistema quando necessário. A última fase é a de testes, utilizada para validar e verificar a eficiência do método proposto, sem a realização de ajustes.

5.2 A escolha da base

A escolha da base de vídeos baseou-se, primeiramente, no seu reconhecimento internacional. Desta forma, buscou-se na Internet os vídeos para a composição da base de testes.

A base *Open Video Project* [24] possui uma grande quantidade de vídeos, porém a maioria são documentários e vídeos educacionais, com poucas edições. Nesta base, também, não são disponibilizados resultados de outras técnicas de segmentação, para que possam ser comparados à abordagem proposta. Assim sendo, estes vídeos foram descartados e não foram incluídos na base de testes deste trabalho.

O *site "Some Results in Video Segmentation*" [8] mostrou-se interessante, pois exibe os resultados de três técnicas de detecção de cortes (*Feature based cut detection with automatic threshold selection*, *Pixel Based Method with Localization* e *Histogram Method Cut Detection)*

e disponibiliza, também, a base de vídeos usada nos testes. Em termos de variedade de tipos de programação, a base corresponde a uma boa amostra, disponibilizando trechos de desenho animado, comercial, *trailer*, filme e seriado. Porém, no aspecto de realidade, esta base não corresponde a amostras reais de programação.

Os vídeos disponibilizados são muito pequenos, a maioria com menos de um minuto, sendo possível verificar que a maioria das amostras de vídeos não possuem outros tipos de transições, apenas cortes. Como pode ser verificado na tabela 5.4, as técnicas de detecção de cortes propostas neste *site* obtiveram resultados muito bons, porém não correspondem a resultados realistas, pois não se deparam com dificuldades e todas as situações que podem existir em uma base realista.

Portanto, buscando um estudo da realidade, sem procurar a facilidade e, tampouco, aumentar a dificuldade, optamos por montar uma base de vídeos mais realista, constituída de cinco jogos de futebol, totalizando mais de 450 minutos de vídeo com as reais dificuldades que um vídeo digital pode apresentar. A base de testes do *site "Some Results in Video Segmentation*" foi, também, utilizada nos testes deste trabalho a fim de se obter um comparativo com outras técnicas.

5.3 Divisão da base

A base de testes utilizada é composta por mais 450 minutos de vídeos de jogos de futebol (cinco jogos) em formato MPEG. Em termos de tamanho, a base supre, satisfatoriamente, as fases de treinamento, validação e testes, pois são aproximadamente oito horas de seqüências de vídeos.

Na análise da base de dados considerou-se, também, a sua relevância em relação ao trabalho e esta mostrou-se satisfatória. A relevância da base pode ser mensurada pela presença das transições que são objeto de estudo do trabalho e de outros tipos de transições, pois estas podem afetar negativamente os resultados de uma detecção. Neste contexto, a base de testes é muito abrangente, pois possui uma vasta coleção de jogos de futebol, apresentando inúmeras transições como cortes, *dissolves*, *fades*, *wipes* e *zooms.*

A base de vídeos foi dividida da seguinte forma: 20% para a utilização no treinamento, 20% para a validação e 60% para a realização dos testes. Portanto, uma amostra de aproximadamente 90 minutos de vídeo corresponde ao conjunto de treinamento. Outra amostra de 90 minutos dos jogos corresponde ao conjunto de validação. Os 270 minutos restantes fazem parte

do conjunto de testes.

5.4 Metodologia de avaliação dos resultados

Após a composição da base de testes, os vídeos foram preparados realizando-se a segmentação *ground-truth*. A segmentação *ground-truth* é uma segmentação ideal, geralmente realizada manualmente, objetivando a avaliação da metodologia proposta.

Neste trabalho, a etapa de segmentação *ground-truth* foi realizada com o auxílio da ferramenta *vidsegpick [6]*, examinado-se visualmente cada vídeo em busca de cortes. Por meio deste programa, é possível percorrer e visualizar o vídeo quadro a quadro, e assim, determinar o ponto exato de uma transição, pois todos os quadros são numerados. Quando um corte é identificado visualmente, o número referente ao quadro que antecede o corte é anotado em um arquivo texto. Cada arquivo texto indica a localização das transições corretas para cada ritmo visual, que devem ser detectadas pelo sistema nas fases de treinamento, testes e validação.

Para possibilitar uma comparação com a segmentação *ground-truth*, um arquivo com a localização dos cortes detectados pelo sistema também é necessário. Desta forma, o sistema percorre a imagem de detecção final buscando a localização de cada corte detectado e anotando-o em um arquivo texto.

A partir dos arquivos com localização dos cortes obtidos pelo sistema e pela segmentação *ground-truth*, uma comparação é realizada, obtendo-se a quantidade de cortes detectados corretamente pela metodologia proposta.

5.5 Medidas de qualidade

De maneira geral, as medidas de qualidade para a segmentação de vídeo analisam a performance na detecção dos eventos buscados no vídeo, conforme as definições a seguir [16]:

Definição 5.1 Verdadeiro Positivo Número de detecções que corretamente correspondem aos eventos buscados no vídeo. Os verdadeiros positivos serão representados por V^+.

Definição 5.2 Falso Positivo Número de detecções que não correspondem ao evento procurado no vídeo. Os falsos positivos serão representado por F^+.

Definição 5.3 Falso Negativo Número de eventos que deveriam ser detectados, porém não foram. Os falsos negativos serão representado por F^-.

Definição 5.4 Verdadeiro Negativo Quando uma amostra negativa (não-ocorrência de um evento) não é detectada. Os verdadeiros negativos serão representados por V^-.

O número de verdadeiros positivos, falsos negativos e falsos positivos são calculados pelo sistema, comparando-se os resultados obtidos entre a segmentação *ground-truth* e a segmentação do sistema.

A partir destas medidas, é possível extrair outras métricas, também, utilizadas na avaliação das técnicas de segmentação de vídeo digital, tais como precisão, revocação e erro, definidas a seguir [16]:

Definição 5.5 Precisão (*Precision*) A taxa de precisão de um algortimo relaciona-se com as corretas e falsas detecções obtidas por um sistema, como definida pela equação 5.1.

$$Precisão = \frac{V^+}{V^+ + F^+} \tag{5.1}$$

Definição 5.6 Revocação (*Recall*) A taxa de revocação de um algoritmo relaciona-se à taxa de corretas detecções, como definida pela equação 5.2.

$$Recall = \frac{V^+}{V^+ + F^-} \tag{5.2}$$

onde $V^+ + F^-$correspondem ao total de eventos que deveriam ser detectados .

Definição 5.7 Erro A taxa de erro de um algoritmo relaciona-se às falsas detecções, como definida pela equação 5.3.

$$Erro = \frac{F^+}{V^+ + F^-} \tag{5.3}$$

5.6 Tempo

É inadequado analisar resultados de uma metodologia sem fornecer ordens relativas a grandeza de tempo. Por este motivo, forneceremos a média de tempo de processamento para testes executados em uma máquina com processador Pentium 4, 3.4 GHz e 1 GB de memória RAM.

Em relação ao tempo de processamento do sistema, este leva, em média, 40 segundos para a montagem do ritmo visual e detecção de bordas em cores de um vídeo de 47 minutos a 30 quadros por segundo (em inglês *frames per seconds–fps*). A parte mais demorada do sistema

corresponde ao processamento de filtragem seqüencial FECABE, limiarização por Abutaleb, limiarização por Bernsen e erosão condicional. Esta etapa do algoritmo demora em média 6 minutos para o mesmo ritmo visual, correspondente a um vídeo de 47 minutos a 30 fps. Portanto, nosso algoritmo leva, em média, o tempo de 0,14 segundos para processar 1 segundo de um vídeo a 30 fps.

Havendo a necessidade de processamento em tempo real, um *buffer* pode ser utilizado. Desta forma, o vídeo pode ser dividido em ritmos visuais de seqüências menores, que podem ser processadas rapidamente.

5.7 Resultados com a base de jogos de futebol

A fase de testes utilizou 60% da base composta de cinco jogos de futebol, totalizando mais de 270 minutos de vídeo. As especificações dos vídeos de jogos de futebol utilizados podem ser verificados na tabela 5.1.

Tabela 5.1: Especificações dos vídeos utilizados

Especificações do Jogos de Futebol				
Vídeo	**Tipo**	**Resolução**	***fps***	**Tempo**
Brasil x Chile - 1°tempo	MPEG1	320x240	30	46’ 54”
Brasil x Chile - 2^o tempo	MPEG1	320x240	30	47’ 42”
Atlético x Botafogo - 1^o tempo	MPEG1	320x240	30	46’ 46”
Atlético x Botafogo- 2^o tempo	MPEG1	320x240	30	48’ 41”
Flamengo x Figueirense - 1^o tempo	MPEG1	320x240	30	46’ 70”
Flamengo x Figueirense - 2^o tempo	MPEG1	320x240	30	47’ 10”

Os resultados apresentados nos testes confirmam a relevância da proposta em estudar a realidade, testando-se a metodologia em uma base de testes realista com todas as dificuldades que um vídeo pode oferecer. Cada vídeo possui em média 47 minutos, totalizando quase 5 horas de vídeo e uma grande quantidade de cortes e outras transições, conforme a tabela 5.2.

A taxa de revocação média de 0.809 revela que a morfologia em cores proposta por Calixto obteve êxito em destacar grande parte dos cortes de difícil detecção presentes na base, onde a diferença de matiz e contraste é mínima.

O valor de precisão médio de 0.78 revela a imensa quantidade de efeitos presentes nesta base, tais como *dissolves, fades, wipes* e *zooms*, pois mesmo com a redução de falsos positivos

Tabela 5.2: Resultados da metodologia proposta aplicada à base de jogos de futebol

Metodologia Proposta						
Vídeo	**Precisão**	**Revocação**	**Erro**	**Cortes**	**Detectados**	**Falsos**
Brasil x Chile - 1°tempo	0.867	0.820	0.125	399	327	50
Brasil x Chile - 2º tempo	0.825	0.805	0.171	369	297	63
Atlético x Botafogo - 1º tempo	0.727	0.889	0.333	198	176	66
Atlético x Botafogo- 2º tempo	0.732	0.830	0.304	194	161	59
Flamengo x Figueirense - 1º tempo	0.800	0.684	0.171	187	128	32
Flamengo x Figueirense - 2º tempo	0.725	0.827	0.313	150	124	47
Média	0.780	0.809	0.236			

da metodologia proposta, falsos cortes foram detectados.

5.8 Resultados com outra base de testes

Para melhor avaliar a metodologia proposta nesta dissertação, surgiu a necessidade de se comparar resultados desta abordagem a resultados de outras metodologias. Portanto, a técnica proposta foi, também, testada com a base de vídeos utilizada no artigo ”*Feature based cut detection with automatic threshold selection*” [7]. Esta base de testes encontra-se disponível no *site ”Some Results in Video Segmentation*” [8], onde há um comparativo de técnicas de segmentação de vídeo.

Esta base de vídeos possui diversos tipos de programação de televisão tais como: desenho animado, trechos de filme colorido, monocromático e branco e preto, seriado de televisão, comercial, noticiário e *trailer* de filme, conforme especificações da tabela 5.3. É importante ressaltar que a maioria dos vídeos são bem pequenos e não contém outros tipos de transições de vídeo. Os resultados fornecidos pelas técnicas do *site ”Some Results in Video Segmentation*” estão descritos na tabela 5.4.

A metodologia proposta pela presente dissertação foi testada nesta mesma base de vídeos e, os resultados encontram-se na tabela 5.5.

Os resultados obtidos com esta base de vídeos não são tão bons quando comparados aos resultados das técnicas do *site*. Os falsos positivos obtidos foram responsáveis pela piora na taxa de precisão em relação ao *site*. Contudo, é necessário esclarecer que não foi possível utilizar toda parte do algoritmo relativa à redução de falsos positivos. O ponto chave na redução de falsos positivos em nossa metodologia é a identificação dos quadros I, mas o formato destes

Tabela 5.3: Especificações dos vídeos utilizados

Especificações dos vídeos				
Vídeo	**Tipo**	**Resolução**	***fps***	**Tempo**
Desenho animado	MPEG1	192x144	30	21”
Trecho de filme	MPEG1	320x142	25	38”
Trecho de filme	MPEG1	384x288	30	53”
Trecho de série de TV	MPEG1	336x272	25	1’ 45”
Trecho de filme	MPEG1	384x288	30	17”
Comercial	MPEG1	384x288	30	16”
Trecho de filme	MPEG1	352x240	25	3’ 25”
Noticiário	MPEG1	384x288	30	15”
Trailer de filme	MPEG1	240x180	23	36”

Tabela 5.4: Tabela comparativa entre abordagens de segmentação de vídeo disponível no *site Some Results in Video Segmentation [8]*

Feature Tracking Method			***Pixel Based Method***		***Histogram MethodCut***	
Vídeo	**Precisão**	**Revocação**	**Precisão**	**Revocação**	**Precisão**	**Revocação**
Desenho animado	1	1	1	1	1	1
Trecho de filme	1	1	0.825	0.825	1	0.325
Trecho de filme	0.595	0.870	0.764	0.778	0.936	0.536
Trecho de série de TV	1	1	1	1	1	0.941
Trecho de filme	0.938	1	0.867	0.867	0.955	0.700
Comercial	0.810	0.944	0.708	0.994	1	0.667
Trecho de filme	0.895	0.895	0.927	1	0.971	0.895
Noticiário	1	1	1	1	1	0.500
Trailer de filme	0.497	0.897	0.623	0.540	0.850	0.395
Média	0.874	0.961	0.774	0.800	0.971	0.701

vídeos são incompatíveis com as funções de manipulação das estruturas do MPEG da biblioteca utilizada. Para melhores resultados, seria preciso a utilização desta parte do algoritmo relativa a identificação dos quadros I, porém, neste momento, não foi possível por falta de tempo.

A disparidade de alguns valores da taxa de revocação destes testes revela a importância da cor em nossa abordagem. As piores taxas de revocação obtidas (0.345 e 0.494) com esta base, referem-se a vídeos muito escuros onde a informação de cor é, praticamente, inexistente. Para vídeos onde a informação de cor é abundante, os valores de revocação obtidos foram bem melhores.

Estes resultados evidenciam que a metodologia proposta busca cortes de difícil detecção

Tabela 5.5: Resultados da metodologia proposta aplicada à base do *site Some Results in Video Segmentation*

Metodologia Proposta					
Vídeo	**Precisão**	**Revocação**	**Cortes**	**Detectados**	**Falsos**
Desenho animado	0.500	0,857	7	6	6
Trecho de filme	0.538	0.875	8	7	6
Trecho de filme	0.836	0.944	54	51	10
Trecho de série de TV	0.872	1	34	34	5
Trecho de filme	1	0.345	29	10	0
Comercial	0.818	1	18	18	4
Trecho de filme	0.882	0,769	39	30	4
Noticiário	0.667	1	4	4	2
Trailer de filme	0.896	0.494	87	43	5
Média	0.779	0.809			

presentes em jogos de futebol, conseqüentemente aumentando os falsos positivos em vídeos de cortes nítidos.

É possível aprimorar a técnica proposta e obter resultados semelhantes aos do site, ajustando os parâmetros do processamento aplicado. No caso do trecho de filmes monocromático e branco e preto, foi necessário aumentar o parâmetro de contraste na Limiarização de Bernsen. Porém, nos demais, optamos por manter a metodologia para detecções de cortes ajustada para a base de testes de jogos de futebol proposta.

Capítulo 6

Análise dos Resultados

6.1 Introdução

A detecção de transições em imagens de jogos de futebol não é das tarefas mais fáceis, pois este tipo de vídeo possui algumas particularidades que dificultam a detecção das transições. Fatores como estes, influenciam nos resultados das taxas de precisão e revocação (seção 5.5), e não podem deixar de ser analisados, pois não existem comentários de como se tratar estes tipos de dificuldades nos demais trabalhos da área.

6.1.1 Diferenças de matiz e contraste

Pelo fato do ambiente de filmagem de jogos de futebol permanecer normalmente o mesmo, praticamente não há diferença de matiz e contraste na presença de cortes, o que dificulta a sua detecção. Contudo, este comportamento é mais raro em outros tipos de programação de vídeo, pois normalmente há uma diferença de ambiente, iluminação e cores na presença de cortes.

Nas imagens de ritmo visual 6.2 e 6.3 na próxima página pode-se perceber claramente a diferença de matiz e contraste na presença de cortes em vídeos de desenho animado e seriado de televisão. Já a imagem 6.1 é um exemplo de um típico ritmo visual criado a partir de um vídeo de jogo de futebol, em que existem três cortes de difícil detecção, devido a pequena diferença de matiz e contraste.

Figura 6.1: Ritmo visual de um jogo de futebol com três cortes de difícil detecção.

Figura 6.2: Ritmo visual de um desenho animado com cortes nítidos.

Figura 6.3: Ritmo visual de uma seriado de televisão com cortes nítidos.

6.1.2 Cortes de difícil detecção

Analisando-se a figura 6.4, pode-se verificar que, apesar da dificuldade em se enxergar a existência de cortes, o processo morfológico baseado na ordenação reduzida e lexicográfica de cores de Calixto os destacou. E, assim a metodologia proposta obteve êxito em localizá-los.

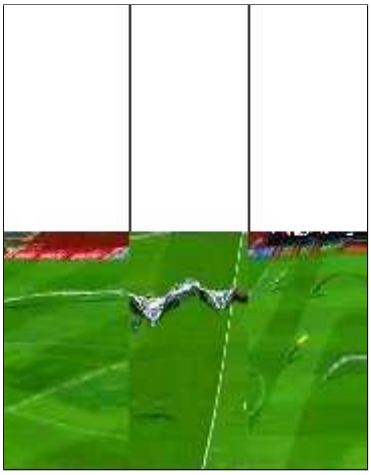

Figura 6.4: Exemplo de corte de difícil detecção.

Na imagem 6.5, existem outros exemplos de cortes de difícil detecção. O processo morfológico baseado na ordenação reduzida e lexicográfica de cores de Calixto conseguiu destacar o corte direito e a metodologia proposta obteve êxito em localizá-lo. Mas, a metodologia proposta não conseguiu identificar o corte esquerdo, pois este foi rejeitado por não tocar a borda superior da imagem.

Figura 6.5: Exemplo de corte de difícil detecção.

A operação de *zoom*, algumas vezes, torna-se um obstáculo na detecção de cortes, pois

as linhas verticais se misturam as imagens de *zoom,* dificultando muito a sua identificação, como exemplificado na figura 6.6, em que a metodologia proposta obteve êxito em identificá-las como corte.

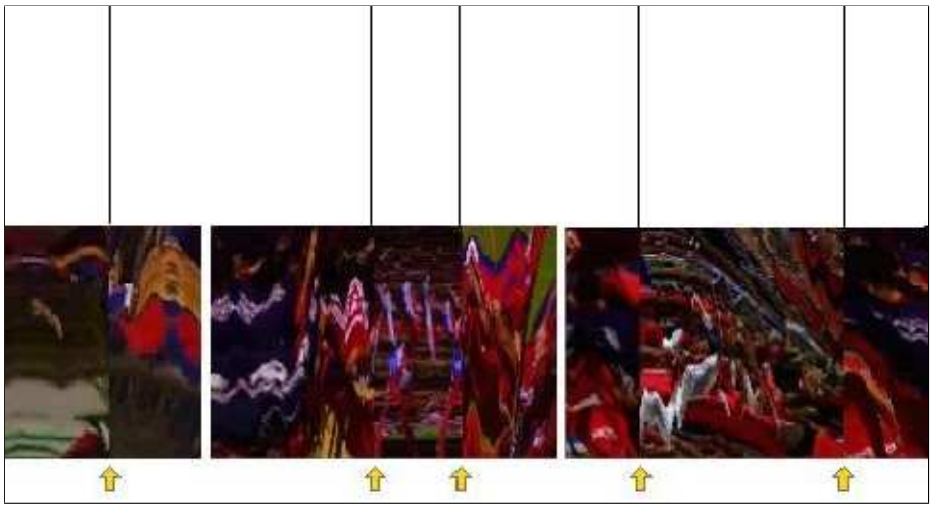

Figura 6.6: Exemplo de cortes de difícil detecção complicados por *zoom*.

6.1.3 Falsos cortes

Os jogos de futebol apresentam, também, algumas peculiaridades que podem inserir uma grande quantidade de falsos positivos. A iluminação natural é responsável pela introdução de linhas na imagem de ritmo visual, devido a diferença de iluminação e grande quantidade de sombras formadas no gramado.

Apesar da sensação visual de se enxergar cortes, como na figura 6.7, o que, de fato, são somente mudanças de matiz verde devido à iluminação do campo, o processo morfológico baseado na ordenação reduzida e lexicográfica de cores de Calixto não as destacou. E, conseqüentemente, a metodologia proposta teve êxito em não localizá-las.

Figura 6.7: Exemplo de sensação de corte por diferença de matiz devido a sombras no campo.

Pelo fato dos jogos de futebol ocorrerem ao ar livre, a diferença de luminosidade cria linhas, que poderiam ser detectadas como cortes, como exemplificado na figura 6.8. Neste exemplo, existem três segmentos de ritmo visual, onde há uma sensação visual de existência de cortes devido às mudanças de matiz verde da iluminação do campo. Porém, o processo morfológico baseado na ordenação reduzida e lexicográfica de cores de Calixto não as destacou. E por conseqüência, a metodologia proposta teve êxito em rejeitá-las como corte.

Figura 6.8: Três exemplos de ritmos em que existe sensação de corte por diferença de matiz devido à iluminação.

6.1.4 Falsos positivos

O grande dinamismo dos jogos é responsável pela inserção de uma grande quantidade de transições graduais, principalmente *dissolves*, necessários para a repetição de jogadas durante uma partida ou a exibição de lances de jogos concorrentes. Estas transições, muitas vezes, geram falsos positivos como exemplificado na figura 6.9.

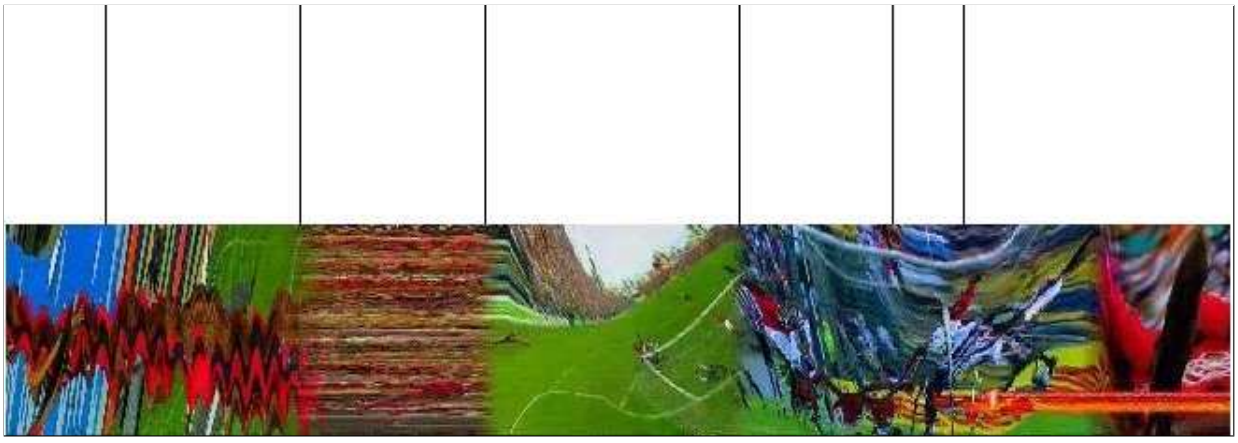

Figura 6.9: Exemplo de inserção de falsos positivos devido aos dissolves.

Outro responsável pela inserção de falsos positivos no ritmo visual é a grande quantidade de operações de *zoom* utilizadas para exibir alguns lances mais proximamente. Esta

conseqüência pode ser verificada na imagens 6.10 e 6.11, onde pode-se observar que algumas linhas verticais são falsos positivos introduzidos pela exibição de lances com *zoom.*

Figura 6.10: Exemplo de falsos positivos provocados por *zoom*.

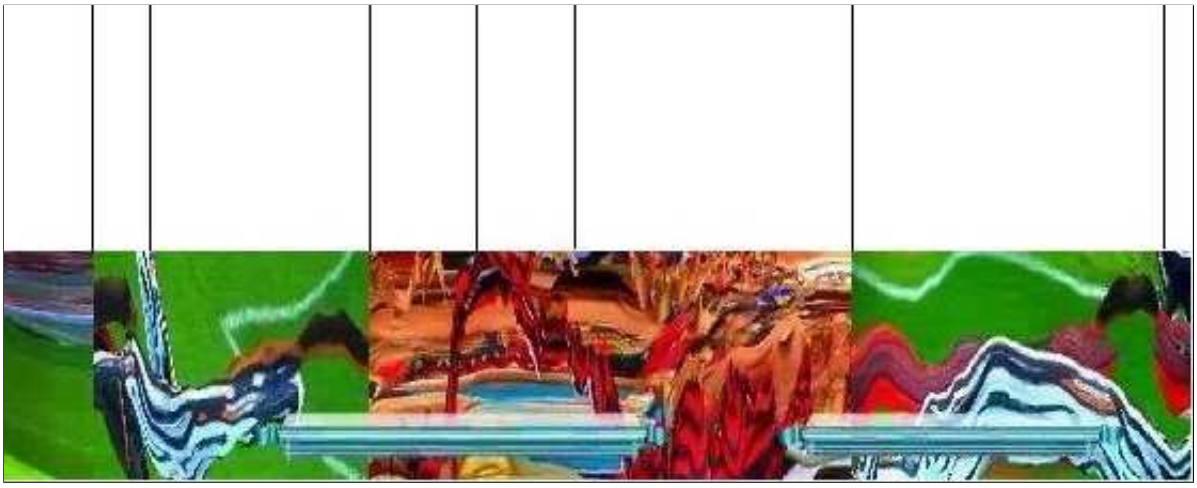

Figura 6.11: Exemplo de falsos positivos provocados por *zoom*.

6.1.5 Definição de fronteira

Existem, também, casos de *dissolves* com *zoom*, nos quais se percebe uma mistura de cores e uma grande dificuldade em se definir computacionalmente uma fronteira, como pode ser observado no exemplo da imagem 6.12 na próxima página. Neste caso, no segundo segmento de ritmo visual, um *dissolve* foi detectado erroneamente como corte.

Figura 6.12: Exemplo *dissolves* com *zoom*.

6.1.6 Linhas verticais de origem desconhecida

Em outras situações, existem linhas verticais no ritmo visual que parecem erros de quantização ou de compactação. Estes casos, também, podem gerar falsos positivos, como exemplificado na imagem 6.13, onde três trechos de ritmo visual apresentam estas verticais. A primeira linha foi detectada erroneamente como corte pela metodologia proposta. A vertical do segundo trecho de ritmo visual foi detectada como corte na primeira etapa do algoritmo, mas foi descartada na segunda fase (etapa de redução de falsos positivos, seção 4.8). Finalmente, a última linha não foi detectada como um corte pela metodologia proposta.

Figura 6.13: Exemplo de linhas de origem desconhecida que parecem cortes.

6.1.7 Cortes ambíguos

Nos exemplos de segmentos de ritmo visual da figura 6.14, é possível observar-se cortes nítidos na parte superior da imagem, mas na borda inferior, as cores são muito semelhantes e não é possível discernir uma linha de corte. Portanto, na detecção de cortes, a linha vertical não se mantém contínua, tocando as bordas superior e inferior da imagem. Nestas situações, é complicado resolver este tipo de ambigüidade e obter uma detecção correta do sistema para todos os casos.

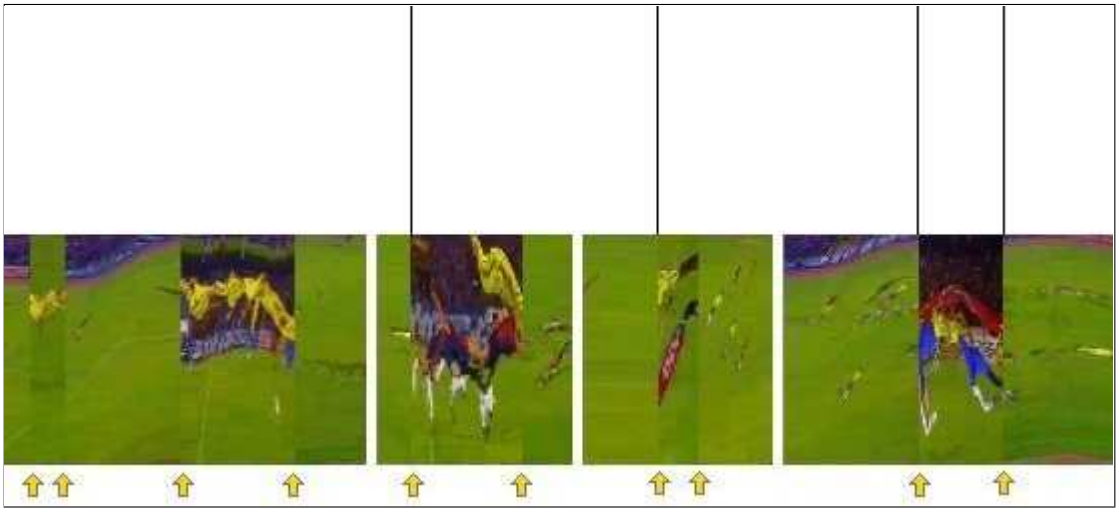

Figura 6.14: Exemplo de cortes de difícil detecção na parte inferior da imagem.

Capítulo 7

Conclusões e Trabalhos Futuros

Nesta dissertação foi apresentada uma nova abordagem para a detecção de cortes em vídeo digital, na qual a análise do vídeo é realizada sobre uma imagem formada a partir da diagonal principal de cada quadro, como na recente abordagem chamada ritmo visual por amostragem. Neste trabalho, duas inovações são introduzidas: o uso da morfologia em cores para detecção das transições e a utilização de vídeo comprimido, fazendo uso das vantagens em se trabalhar diretamente no domínio MPEG.

Com a utilização da morfologia em cores para a detecção de transições, a preciosa informação de cor não é descartada como em outras abordagens, possibilitando difíceis detecções onde esta informação extra é importante. O ritmo visual colorido é criado rapidamente, sem a necessidade da prévia conversão para níveis de cinza, que apenas é utilizada quando a quantidade de informações é muito menor.

Trabalhando-se diretamente sobre vídeos comprimidos e dispensando-se a sua descompressão, obtêm-se maior rapidez devido à grande disponibilidade de vídeos digitais em formato comprimido. Utilizando-se miniaturas de quadros no domínio MPEG, um ritmo visual menor é construído, diminuindo-se, consideravelmente, a quantidade de informação e, conseqüentemente, o tempo de processamento, possibilitando o uso desta técnica em sistemas em tempo real.

A taxa de revocação média de 0.809, obtida com os testes aplicados à base de jogos de futebol, evidencia que, mesmo com a grande quantidade de cortes de difícil detecção, em que a diferença de matiz e contraste é mínima, a morfologia em cores proposta por Calixto conseguiu destacar grande parte destas complexas transições. Desta forma, um comparativo com outras abordagens de morfologia em cores faz-se necessário, a fim de verificar o comportamento de

diferentes ordenações de cores com estes cortes de difícil detecção.

O valor de precisão médio de 0.78, obtido com os testes realizados com a base de jogos, mostra a real complexidade da base de testes utilizada, pois mesmo com a aplicação da etapa do sistema relativa à redução de falsos positivos, ainda assim, muitos falsos cortes foram detectados. Este comportamento revela a imensa quantidade de efeitos diferentes presentes nesta base, tais como *dissolves*, *fades*, *wipes* e *zooms*. Revela, também, que ao se tentar destacar os cortes mais difíceis, que exigem um processamento mais abrangente, aumenta-se o número de falsas detecções.

Nesta pesquisa, os testes realizados utilizaram os mesmo parâmetros nas técnicas aplicadas aos diferentes tipos de vídeos. Porém, cada tipo de vídeo possui uma dificuldade de detecção diferente. Assim, melhores resultados podem ser obtidos através do ajuste de parâmetros das técnicas utilizadas para cada vídeo, como, por exemplo, o número de iterações na etapa de filtragem, o valor do contraste ou janela na limiarização local. Para a realização deste ajuste de parâmetros, é possível implementar uma etapa de calibração do sistema, em que o primeiro minuto do vídeo define se os parâmetros devem ser ajustados para uma detecção mais complexa ou mais simples.

É necessário destacar, ainda, que obtivemos os melhores resultados de precisão para os vídeos que possuem as maiores quantidades de cortes. Esta informação revela que os vídeos com menos cortes possuem maior número de outras transições como *dissolves*, *fades* e *wipes*, que aumentam o número de falsos positivos e, conseqüentemente, diminuem a taxa de precisão.

Os resultados obtidos são importantes a medida que revelam um estudo da realidade, aplicando uma metodologia voltada a uma base de testes com todas as dificuldades que um vídeo real pode oferecer. Desta forma, faz-se necessária a obtenção de comparativos de outras metologias com a mesma base de jogos de futebol, visando uma melhor avaliação dos resultados.

Entre os possíveis trabalhos a serem realizados futuramente, destacam-se:

- A implementação de outros métodos de segmentação de vídeo aplicados a esta base de jogos de futebol é um importante trabalho para obtenção de um real comparativo.

- Realizar os testes de detecção de cortes para ritmos visuais criados a partir de miniaturas de 1/8 de quadro, visando um comparativo de tempo e precisão entre os resultados obtidos nesta pesquisa e resultados obtidos com imagens DC menores.

- A implementação desta metodologia utilizando outras técnicas de morfologia em cores a fim de se obter um comparativo de eficiência nas detecções.
- Um novo estudo das etapas de processamento para a detecção dos cortes é necessário, objetivando aumentar a taxa de acerto das detecções.
- Diminuir alguns passos de processamento, visando simplificar ainda mais a metodologia e reduzir o tempo de processamento.
- Realizar a detecção de cortes a partir da combinação de informações relativas a amostras horizontal, vertical e diagonal de cada quadro.
- Desenvolver uma nova metodologia que realiza as detecções a partir do uso de um classificador de padrões de cortes aplicado ao ritmo visual.

Referências Bibliográficas

[1] Akutsu, A., and Tonomura, Y. Video tomography: An efficient method for camerawork extraction and motion analysis. In *ACM Multimedia* (1994), pp. 349–356.

[2] Alvarenga, V. A., Infantosi, A. F., Azevedo, C. M., and Pereira, W. C. A. Aplicação de operadores morfológicos na segmentação e determinação do contorno de tumores de mama em imagens por ultra-som. *SBEB* (2003), 91–101.

[3] Arman, F., Hsu, A., and Chiu, M.-Y. Image processing on compressed data for large video databases. In *ACM Multimedia '93* (Anaheim, California, 1993), P. V. Rangan, Ed., ACM Press, pp. 267–272.

[4] Bernsen, J. Dynamic thresholding of gray-level images. In *Proc. Eighth Int'l Conf. on Pattern Recognition* (Paris - France, 1986), pp. 1251–1255.

[5] Bortoleto, C. M. Multicast semi-confiável para aplicações multimídia distribuídas. Master's thesis, Pontifícia Universidade Católica do Paraná, Brasil, 2005.

[6] Bose, P., Laganiere, R., and Whitehead, A. Vidsegpick, 2003. Disponível em http://vision.scs.carleton.ca/ awhitehe/vidproc/. Acessado em: 29 de junho de 2007.

[7] Bose, P., Laganiere, R., and Whitehead, A. Feature based cut detection with automatic threshold selection. *Int. Conf. on Image and Video Retrieval* (2004), 410–418.

[8] Bose, P., Laganiere, R., and Whitehead, A. Some results in video segmentation, 2004. Disponível em http://www.site.uottawa.ca/ laganier/videoseg. Acessado em: 10 de junho de 2007.

[9] Calixto, E. Granulometria morfologica em espaços de cores: estudo da ordenaçao espacial. Master's thesis, Universidade Federal Fluminense, Agosto 2005.

[10] Cheriet, M., Said, J. N., and Suen, C. Y. A recursive thresholding technique for image segmentation. *IEEE Transactions on Image Processing 7*, 6 (1998), 918–921.

[11] Chun, S. S., Kim, H., Kim, J.-R., Oh, S., and Sull, S. Fast text caption localization on video using visual rhythm. In *VISUAL* (2002), pp. 259–268.

[12] Davenport, G., Smith, T. A., and Pincever, N. Cinematic primitives for multimedia. *IEEE Comput. Graph. Appl. 11*, 4 (1991), 67–74.

[13] Efg's Computer Lab. Hsv, 2005. Disponível em: http://www.efg2.com/Lab/Graphics/Colors/HSV.htm. Acessado em: 09 de abril de 2006.

[14] Facon, J. *Morfologia Matemática: Teoria e exemplos*. Editora Universitária Champagnat da Pontifícia Universidade Católica do Paraná, Curitiba, Brasil, 1996.

[15] Feng, J., Lo, K., and Mehrpour, H. Scene change detection algorithm for mpeg video sequence. In *International Conference on Image Processing (ICIP'96)* (Lausanne, 1996).

[16] Guimarães, S. J. F. *Video transition identification based on 2D image analysis*. PhD thesis, Universidade Federal de Minas Gerais, March 2003.

[17] Hampapur, A., Jain, R., and Weymouth, T. E. Production model based digital video segmentation. *Multimedia Tools Appl. 1*, 1 (1995), 9–46.

[18] Kapur, J., Sahoo, P., and A.K.C.Wong. A new method ofr gray-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics and Image Processing 41* (1985), 273–285.

[19] Kasturi, R., and Jain, R. C. *Computer Vision: Principles*. IEEE Computer Society Press, Los Alamitos, CA, 1991.

[20] Kim, H., Lee, J., Yang, J.-H., Sull, S., Kim, W. M., and Song, S. M.-H. Visual rhythm and shot verification. *Multimedia Tools Appl. 15*, 3 (2001), 227–245.

[21] Koprinska, I., and Carrato, S. Detecting and classifying video shot boundaries in mpeg compressed sequences. In *IX European Signal Processing Conference (EUSIPCO)* (Rhodes, 1998), pp. 300–304.

[22] Koprinska, I., and Carrato, S. Temporal video segmentation: A survey. *Signal Processing: Image Communication 16* (2001), 477–500.

[23] Lefèvre, S., Holler, J., and Vincent, N. A review of real-time segmentation of uncompressed video sequences for content-based search and retrieval. *Real-Time Imaging 9*, 1 (2003), 73–98.

[24] Marchionini, G., and Geisler, G. Open video project, a shared digital video collection. Disponível em http://www.open-video.org. Acessado em: 10 de junho de 2007.

[25] Mattana, M. F., and Facon, J. Avaliação por reconhecimento da qualidade da segmentação por binarização de cheques bancários. Master's thesis, Pontifícia Universidade Católica do Paraná, Brazil, August 1999.

[26] Meng, J., Juan, Y., and Chang, S. F. Scene change detection in a mpeg compressed video sequence. In *IS&T/SPIE International Symposium on Eletronic Imaging* (San Jose, 1995), vol. 2417, pp. 14–25.

[27] Milstein, N. Image segmentation by adaptative thersholding. *Technion Israel Institute of Technology* (1998), 33.

[28] Nagasaka, A., and Tanaka, Y. *Visual Database Systems ll*. Elsevier Science Publishers B. V., North-Holland, 1992, pp. 113–127.

[29] Ngo, C. W. *Analysis of spatio-temporal slices for video content representation*. PhD thesis, Hong Kong University of Science and Technology, August 2000.

[30] OǴorman, L., and Kasturi, R. *Document Image Analysis*. IEEE Computer Society Press, January 1995, ch. Pixel-level Processing, pp. 7–12.

[31] Otsu, N. A threshold selection method from grey-level histograms. *IEEE Transactions in Systems, Man and Cybernetics 9* (January 1979), 62–66.

[32] Pappas, T. N., and Jayant, N. S. An adaptive clustering algorithm for image segmentation. In *Second International Conference on Computer Vision* (Tampa, FL, 1988), IEEE Computer Society, pp. 310–315.

[33] Pereira, G. A. S., and Yehia, H. C. Mpeg-2 um estudo do padrão de vídeo, 1999. Disponível em http://homepages.dcc.ufmg.br/ gpereira/mpeg/mpeg.html. Acessado em: 22 de janeiro de 2006.

[34] Ritter, G. X., and Wilson, J. *Handbook of Computer Vision Algorithms in Image Algebra*. CRC Press, North Carolina, 1996.

[35] Sahoo, P., Soltani, S., and A.K.C.Wong. A survey of thresholding techniques. *Computer Vision, Graphics and Image Processing 41* (1988), 233–260.

[36] Sahoo, P. K., Wilkins, C., and Yeager, J. Threshold selection using renyi's entropy. *Pattern Recognition 30*, 1 (1997), 71–84.

[37] Scherrer, R. Vídeo na internet, 2002. Disponível em http://www.ead.unifei.edu.br/public/artigos.htm. Acessado em: 07 de janeiro de 2006.

[38] Shen, K., and Delp, E. A fast algorithm for video parsing using mpeg compressed sequences. In *International Conference on Image Precessing ICIP'96* (Lausanne, 1996).

[39] Souza, A. I., and Santos, C. A. Morfologia matemática, 1998. Disponível em http://www.inf.ufsc.br/ visao/morfologia.pdf. Acessado em: 29 de abril de 2007.

[40] Sundaram, H. *Segmentation, Structure Detection and Summarization of Multimedia Sequences*. PhD thesis, Columbia University, 2002.

[41] UC Berkeley Multimedia Research Center, 2006. Disponível em http://bmrc.berkeley.edu/frame/research/mpeg. Acessado em: 22 de fevereiro de 2006.

[42] Viapiana, E. L., and Engel, P. M. Um estudo comparativo de aplicação de técnicas de segmentação em imagens reais. Anais da Semana Acadêmica do CPGCC da Universidade Federal do Rio Grande do Sul, 1998.

[43] Xiong, W., Lee, J., and Ip, M. Net comparison: a fast and effective method for classifying image sequences. In *SPIE Conference on Storage and Retrieval for Image and Video Database III* (San Jose, CA, 1995), vol. 2420, pp. 318–328.

[44] Yeo, B., and Liu, B. Rapid scene analysis on compressed video. *IEEE Trans. Circuits Systems Video Technol. 5*, 6 (1995), 533–544.

[45] Zabih, R., Miller, J., and Mai, K. A feature-based algorithm for detecting and classifying production effects. *Multimedia Systems 7*, 2 (1999), 119–128.

[46] Zhang, H., Kankanhalli, A., and Smoliar, S. W. Automatic partitioning of full-motion video. *Multimedia Syst. 1*, 1 (1993), 10–28.

[47] Zhang, H., Low, C. Y., and Smoliar, S. W. Video parsing and browsing using compressed data. *Multimedia Tools Appl. 1*, 1 (1995), 89–111.

[48] Zhang, H. J., Low, C. Y., Gong, Y. H., and Smoliar, S. W. Video parsing using compressed data. In *SPIE Conference on Image and Video Processing II* (1994), pp. 142–149.