MÁRCIO LUIZ ROSSATO GOMES

RECUPERAÇÃO DE VÍDEOS POR CONTEÚDO COM BASE EM INFORMAÇÕES ESTÁTICAS E DINÂMICAS

Dissertação de Mestrado apresentada ao Curso de Pós Graduação de Informática Aplicada da Pontifícia Universidade Católica do Paraná do Campus de Curitiba, como requisito para obtenção do título de Mestre em Informática Aplicada.

CURITIBA
2006

MÁRCIO LUIZ ROSSATO GOMES

RECUPERAÇÃO DE VÍDEOS POR CONTEÚDO COM BASE EM INFORMAÇÕES ESTÁTICAS E DINÂMICAS

Dissertação de Mestrado apresentada ao Curso de Pós Graduação de Informática Aplicada da Pontifícia Universidade Católica do Paraná do Campus de Curitiba, como requisito para obtenção do título de Mestre em Informática Aplicada.

Área de Concentração: Visão, Imagem e Robótica

Orientador: Prof. Dr. Jacques Facon
Co-orientador: Prof. Dr. Alceu de Souza Britto Junior

CURITIBA
2006

DEDICATÓRIA

Aos meus pais, exemplo de vida e profissão, que sempre estiveram ao meu lado em todos os momentos de minha vida e foram responsáveis por tornar meus sonhos em realidade.

A minha esposa Daiane pela grande paciência e por ter sido compreensiva nos momentos difíceis que surgiram durante esta jornada.

Ao meu tio Marcos, por ter incentivado o inicio desta jornada tão longe de minha terra natal.

AGRADECIMENTOS

Aos Prof. Dr. Alceu de Souza Brito Junior, e Prof. Dr. Jacques Facon,

Meus incentivadores, guias e mestres sempre atentos, pelo estímulo, pela paciência e pelas inúmeras orientações sem as quais este trabalho nunca teria sido possível.

“Se cada dia cai, dentro de cada noite, há um poço onde a claridade está presa há que sentar-se na beira do poço da sombra e pescar luz caída com paciência”.

Pablo Neruda (Últimos Sonetos)

SUMÁRIO

LISTA DE FIGURAS

LISTA DE SIGLAS

AVI .....Audio Video Interleaved

CBIRS ..... Content Based Image Retrieval Systems

CD ..... Compact Disk

CIE .....Commission International de l´Eclairage

CODEC ..... Coder/Decoder

DCT..... Discrete Cosine Trasnform

DVD.....Digital Video Disc

ECR..... Taxa de Mudança de Quadro

FPS .....Frames Per Second

HSV.....Hue, Saturation, Value, Hue, Saturation, Value

JPEG.....Joint Photografic Experts Group

MHZ ..... Mega Hertz

MPEG.....Moving Pictures Expert Group

NTSC .....National Television Standards Committee

PAL ..... Phase Alternate Line

RGB .....Red, Green, Blue

RVBC .....Recuperação de Video Baseada em Conteúdo

SAD.....Soma das Diferenças Absolutas

SRBC .....Sistema de Recuperação de Vídeo Baseado em Conteúdo

SRI .....Sistema de Recuperação de Informação

SRVBC.....Sistema de Recuperação de Vídeo Baseado em Conteúdo

VCD..... Video Compact Disc

XML..... eXtensible Markup Language

RESUMO

Este trabalho apresenta um método de indexação para recuperação de vídeos que leva em consideração a extração de características de cor e textura, combinados em um espaço *wavelet* juntamente com características de movimento representadas pelo fluxo ótico. Isto é realizado com a inclusão de técnicas de recuperação de informação que permitem tratar os vídeos de forma natural. A escolha por estas características foi atribuída ao fato da cor representar a imagem como um todo, proporcionando informações relevantes para a tarefa de recuperação. Já a característica textura foi escolhida por estar intimamente relacionada com a característica cor, a qual é tratada em diferentes níveis de multi-resolução. O movimento por sua vez, é utilizado pelo fato do mesmo ser capaz de distinguir certas cenas que apesar de visualmente semelhantes possuem conteúdos muito distintos devido a sua dinâmica, como o caso de uma cena de um piquenique no campo e um jogo de futebol, por exemplo.

Palavras Chaves: Recuperação de Vídeo por Conteúdo, Wavelets, Fluxo Ótico.

ABSTRACT

This work presents a indexing method to video retrieval that take in consideration the color and texture features extration combined in a wavelet space together with motion features represented by the optical flow. This is carried through with the inclusion of techniques of information retrieval that allow to deal with the natural videos form. The choice for these features was attributed to the fact of the color to represent the image as a whole, providing excellent information for the retrieval task. Already the texture feature was chosen because is related with the color feature, which is treated in different levels of multiresolution. The motion, is used to distinguish certain scenes that although visually similar possess distinct motion contents like a scene of a picnic and a football match, for example.

Keywords: Content Based Vídeo Retrieval, Wavelets, Optical Flow

1 INTRODUÇÃO

Computadores mais rápidos e baratos, dispositivos com alta capacidade de armazenamento, conexões mais rápidas com a Internet e serviços que propiciam o compartilhamento de informações são alguns dos fatores que têm sido o combustível para a alta demanda de soluções que permitam a edição, reprodução, organização e compartilhamento de vídeos digitais.

Atualmente o usuário de um computador dispõe de ferramentas capazes de reproduzir e editar vídeos digitais com a mesma facilidade que realiza tarefas comuns, como a edição de textos e a navegação na Internet. Este pode produzir vídeos de excelente qualidade sem a necessidade de grandes conhecimentos técnicos e com um consumo de tempo muito pequeno. Um exemplo da crença no potencial da popularização dos vídeos digitais é a inclusão do *Movie Maker* (Microsoft, 2004) nas versões do *Windows ME* (Microsoft, 2000) e *Windows XP* (Microsoft, 2001). Esta ferramenta é um editor de vídeos simples, mas capaz de realizar praticamente todas as tarefas de edição.

Neste contexto, o problema de indexação consiste em que apesar da grande quantidade de imagens e vídeo em forma digital disponíveis atualmente, ainda não temos ferramentas adequadas para, a partir de uma amostra trazer ao usuário itens similares a esta amostra e de interesse do usuário de forma satisfatória.

Recuperar vídeos não implica em devolver ao usuário vídeos com conteúdos idênticos a uma referência fornecida por esse, mas sim documentos que possuam alguma similaridade de conteúdo.

A escassez de ferramentas com esta finalidade pode ser atribuída, à grande dificuldade encontrada em se definir um conjunto de características capaz de representem de forma adequada os vídeos. Como conseqüência desta dificuldade em

representar as características de um vídeo torna-se bastante complexa a tarefa de comparar ou estabelecer semelhanças entre trechos de filmes.

A complexidade implícita na definição de similaridade torna bastante difícil organizar e indexar vídeos de bibliotecas digitais. Com a indexação e organização dificultada pela dificuldade encontrada em se executar estas tarefas, a recuperação de seqüência de vídeos fica bastante comprometida.

1.1 OBJETIVOS

Desta forma, o objetivo principal desta pesquisa é desenvolver uma metodologia de indexação de vídeos baseada em informações estáticas (cor e textura) e dinâmicas (fluxo ótico). Bem como propor uma forma de aplicar estas características combinadas para avaliar seu impacto na recuperação de vídeos por conteúdo. Para se atingir este objetivo serão cumpridas as seguintes etapas:

- Levantamento bibliográfico sobre as técnicas de indexação existentes na literatura;
- Implementação de uma metodologia de segmentação de vídeos;
- Implementação de métodos para extração e seleção de quadros chaves;
- Adaptação e extração das características estáticas propostas por Ramos et al. (2005);
- Implementação e extração de características dinâmicas baseadas em fluxo ótico;
- Criação e rotulação de uma base de vídeos;
- Avaliação das características estáticas e dinâmicas na indexação de vídeos;

1.2 MOTIVAÇÃO

A multimídia têm crescido muito nos últimos anos, com isso as bibliotecas digitais se tornaram populares e vêm sendo vistas como um componente muito importante. As imagens e vídeos são importantes para as mais variadas áreas, seja medicina, meteorologia ou apenas para lazer. Devido a esta importância, muitos esforços têm sido realizados com o objetivo de fornecer soluções para a indexação de vídeos, basicamente os métodos encontrados na literatura têm em comum a segmentação do vídeo digital em unidades menores e extração de alguns quadros para serem analisados da mesma forma que se analisa e recupera imagens estáticas.

Apesar de existir na literatura diversos métodos para se definir conjuntos de características a partir das informações intrínsecas de imagens e vídeos, nenhum deles se mostrou totalmente eficaz. Certas características podem ser bastante significativas em alguns casos bem como apenas atrapalhar em outros, apesar dos inúmeros esforços voltados para se resolver esta questão um conjunto de características que seja eficiente em todos os casos ainda é uma questão em aberto.

Com esse trabalho pretende-se analisar o conjunto de características extraído dos quadros chaves, e desta analise pretende-se avaliar o impacto do movimento na indexação de vídeos. Esta avaliação tem como intenção de contribuir para possibilitar a organização e recuperação mais rápida das bibliotecas digitais.

1.3 PROPOSTA

Desta forma, a proposta deste trabalho é apresentar uma metodologia de indexação de vídeos baseada em informações estáticas (cor e textura) e dinâmicas (fluxo ótico) e aplicar estas características combinadas para avaliar seu impacto na recuperação de vídeos por conteúdo.

A extração das características estáticas pode ser feita de várias formas, mas neste trabalho a extração de características se divide em dois tipos de características: estáticas e dinâmicas. As características estáticas são extraídas focando-se em cor e textura através de wavelets no espaço de cor *HSV* conforme o trabalho de Ramos et al. (2005). Já as características dinâmicas são representadas pela estimativa do fluxo ótico do quadro chave em relação aos quadros próximos a este na tomada. Para uma melhor representação dos atributos de movimento neste trabalho são feitos experimentos para determinar a melhor forma de extrair as características do fluxo ótico.

Também neste trabalho ambos atributos são devidamente ponderados para que seja possível determinar a influência destes atributos na recuperação e encontrar a combinação que traz melhores resultados para a recuperação de vídeos por conteúdo. A proposta deste trabalho é apresentar uma metodologia de indexação de vídeos baseada em informações estáticas (cor e textura) e dinâmicas (fluxo ótico) e aplicar estas características combinadas para avaliar seu impacto na recuperação de vídeos por conteúdo.

Inicialmente um vídeo de referência é submetido a um processo de segmentação de tomadas. Para cada tomada encontrada é feita uma seleção de quadros chaves para representá-la. Em seguida os quadros são analisados para que das informações estáticas (cor e textura) e dinâmicas (movimento), seja criada uma assinatura de características e essa assinatura que é salva em uma base de dados.

Os quadros chaves têm sua assinatura usada na etapa de recuperação propriamente dita, onde as assinaturas que foram obtidas são comparadas e recebem um *score* de acordo com o seu grau de semelhança. A Figura 1.1 ilustra o método proposto.

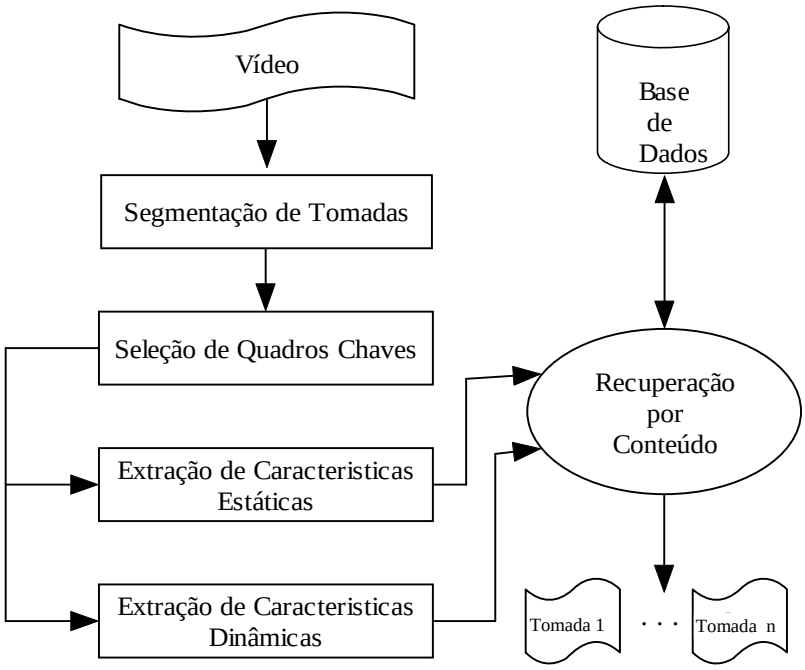


Figura 1.1 - Estrutura da Metodologia Proposta

1.4 CONTRIBUIÇÕES

Dentre as contribuições deste trabalho podemos citar:

- A criação de uma base de vídeos rotulados: Esta base por ter as tomadas e capítulos rotulados permite seu uso em trabalhos futuros tanto na área de recuperação de vídeo por conteúdo como de analise de semântica entre outros.

- A adaptação das características estáticas: Estas características podem ser usadas em filmes e ter sua eficiência avaliada para seqüências de vídeo.
- Avaliação de características dinâmicas: As características permitem analisar o impacto do movimento na indexação de vídeos.
- Avaliação da combinação das características estáticas e dinâmicas na indexação de vídeos.
- Criação de um *framework:* Este *framework* permite com que outros trabalhos possam com maior rapidez detectar tomadas, extrair as características aqui propostas e calcular similaridade entre imagens e quadros de vídeos.

Com isso pretende-se avaliar o uso dos atributos de cor, textura e movimento separadamente e combinados, neste espaço de representação para recuperação de vídeos. Este trabalho também pretende ajudar a suprir a necessidade ferramentas capazes de manipular, organizar e recuperar de forma eficiente vídeos digitais.

1.5 ORGANIZAÇÃO

Após essa breve introdução no Capítulo 1, no Capítulo 2 apresenta um levantamento sobre os Sistemas de Recuperação de Vídeo Baseado em Conteúdo (SRVBC), onde enfatizam-se os pontos principais de tais sistemas. Em seguida o Capítulo 3 descreve o método proposto, bem como sua arquitetura geral. Descreve-se desde a segmentação de vídeo até a fase de recuperação passando por todas as etapas intermediarias. Após isso, o Capítulo 4 mostra os experimentos realizados com

intuito de avaliar o uso de características estáticas concomitantemente com as de dinâmicas. As conclusões gerais e linhas de futuras pesquisas são delineadas no Capítulo 5 deste trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Os sistemas de recuperação de vídeos partem do principio da recuperação de informação. Calvin Mooers nos dá uma boa definição sobre a recuperação de informação:

> “A recuperação de informação trata dos aspectos intelectuais da descrição da informação e sua especificação para busca e também de qualquer sistema técnicas ou máquinas que são empregadas para realizar esta operação”.(MOOERS apud Ferneda, 2003, p. 21)’

A recuperação de informação pode ser designada numa forma mais ampla ao subordinar a mesma ao tratamento de informação (catalogação, indexação e classificação). O termo também pode ser empregado para designar a operação que fornece uma resposta mais ou menos elaborada a uma demanda (Ferneda, 2003).

Este processo é a base da recuperação de vídeos por conteúdo, a qual também consiste em identificar no conjunto da base de dados quais destes atendem a necessidade do usuário.

Um sistema de recuperação de informação (SRI) pode ser estruturado conforme a Figura 2.1.

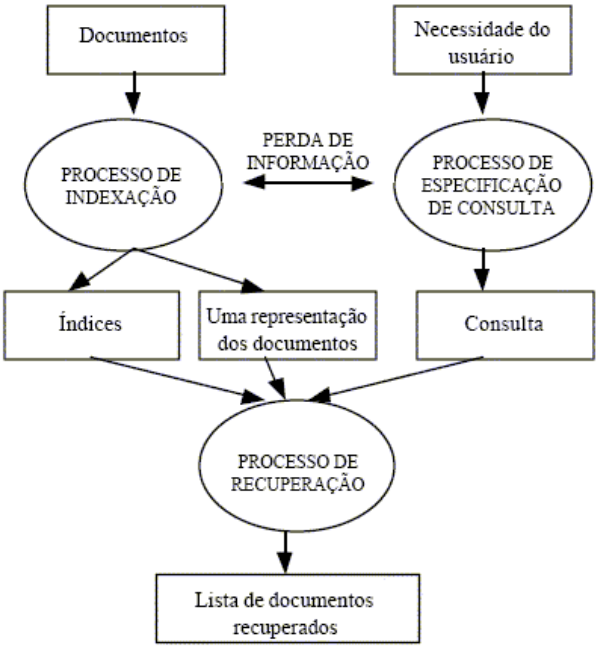


Figura 2.1 - Componentes de um SRI. (Cardoso, 2000)

Conforme Wikipedia (2006), a recuperação de vídeo baseada em conteúdo (RVBC) é uma aplicação da visão computacional para o problema de recuperação de vídeos. “Baseada em Conteúdo” significa que a procura faz uso dos conteúdos dos próprios vídeos, o que é melhor e mais rápido que confiar em meta dados (como títulos ou palavras chaves) informados por pessoas.

O grande interesse na área de RVBC é causado principalmente pelas limitações existentes nos sistemas baseados em meta dados. Informações textuais podem ser facilmente recuperadas através da tecnologia atual, mas há a necessidade da cena ser descrita por um humano o que é impraticável em grandes quantidades de vídeos ou em vídeos que são transmitidos ao vivo sem falar dos problemas de ambigüidade ou subjetividade decorrentes das descrições manuais.

Em Nebulasearch (2005), diz-se que um sistema de recuperação de vídeo por conteúdo (SRVBC) é um sistema que aplica a RVBC. O SRVBC visa evitar a descrição textual e recuperar vídeos tendo como base apenas suas características visuais, as quais devem ser extraídas, armazenadas e para sua recuperação, comparadas. Feito

isso é dada a recuperação caso a tomada analisada contenha semelhança com a tomada de origem conforme exemplifica a Figura 2.2.

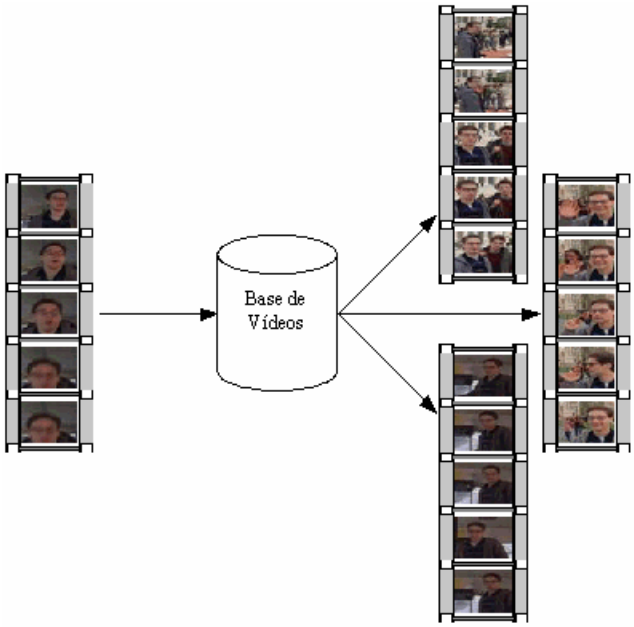


Figura 2.2 - Esquema de um Sistema de Recuperação de Vídeos por Conteúdo

No mesmo sentido, os sistemas de recuperação de imagem baseados em conteúdo (*CBIRS*) quando aplicados às imagens, devem organizar as informações relevantes, e permitir a recuperação de imagens de forma eficiente e conforme a necessidade do usuário. (Ramos et al., 2005)

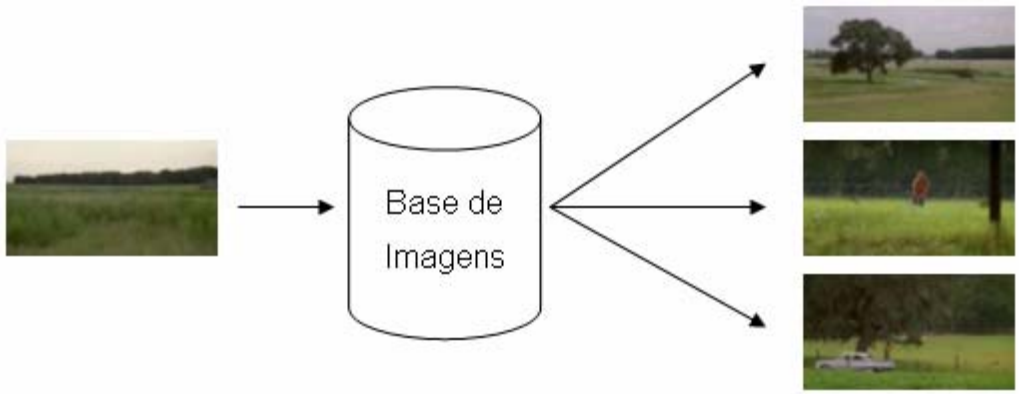


Figura 2.3 – Esquema de um Sistema de Recuperação de Imagens por Conteúdo

A definição dos sistemas de recuperação de vídeo por conteúdo e seu objetivo nos são dadas por Maillet (2002), o qual afirma que podemos definir como objetivo dos *SRVBC* assistir ao operador humano (usuário) recuperar uma seqüência de vídeo (alvo) em uma grande banco de vídeos.

Desta forma podemos concluir que os sistemas de recuperação de vídeos por conteúdo são uma extensão natural da recuperação de imagens por conteúdo, pois, para evitar a descrição textual os SRVBC usam quadros que nada mais são além de imagens.

A recuperação de imagens é feita a partir de informações pictoriais. Isto é possível através da busca de imagens semelhantes a uma imagem de referência por meio de medidas de similaridade (GUDIVANA e RAGHAVAN apud Cardoso, 2000). Para isso, são desenvolvidos métodos que calculam a diferença entre as imagens que estão sendo comparadas. Atualmente, esses métodos utilizam cor, textura e forma como atributos de indexação, os quais são extraídos de maneira independente.

2.1 VÍDEOS DIGITAIS

Um vídeo normalmente é a combinação de 25 a 30 quadros de imagens gravadas por segundo, criando assim a sensação de movimento. Nestas imagens podemos adicionar som o que pode tornar esse vídeo mais interessante e atrativo.

A diferença básica entre o vídeo analógico e o digital consiste na sua forma. Os vídeos analógicos são vídeos basicamente “crus”. Esses vídeos costumam ser armazenados em fitas magnéticas que se degradam com o tempo.

Vídeos digitais possuem a linguagem dos computadores, ou seja, números binários (“0” s e “1” s) que combinados podem descrever as cores e o brilho do vídeo.

Desta forma o termo “vídeo digital” se aplica aos vídeos que empregam a tecnologia digital ao invés de técnicas analógicas.

As vantagens do vídeo digital sobre o vídeo analógico são similares às oferecidas pelas tecnologias de áudio digital. Os vídeos digitais podem ser vistos, editados e copiados sem nenhuma perda de qualidade. Propriedades extras como musica, títulos, legendas e efeitos especiais são muito mais simples de serem inseridos nos vídeos digitais, enquanto a produção de vídeos analógicos e edição são processos muito mais complexos necessitando muitas vezes conhecimentos e equipamentos profissionais para a obtenção de resultados satisfatórios. Além disso, os vídeos digitais podem ser distribuídos em várias mídias como a Internet, DVDs e CDs.

2.1.1 Captura

Em câmeras digitais, com a capacidade de criar vídeos digitais, o sinal é digitalizado e comprimido a medida em que se filma. Com o uso de um computador esse vídeo pode ser facilmente transferido da câmera.

Uma alternativa na produção de vídeos digitais é usar uma câmera de captura de vídeo analógico e um computado munido de uma placa de captura de vídeo. Neste caso o processo de digitalização do vídeo acontece no computador e não na câmera.

2.1.2 Conteúdo Estrutural

Vídeos digitais possuem uma estrutura sintática (apesar de implícita) composta de uma hierarquia de componentes como cenas e tomadas.

Cada vídeo, quando visto em seu nível mais alto possui várias cenas, as cenas são formadas por uma ou mais unidades menores denominadas de tomadas. A tomada por sua vez é composta por quadros com conteúdo suave e contínuo. Esta estrutura pode ser observada na Figura 2.4.

Uma tomada ou *shot* é uma seqüência de quadros inquebrável, tomados de uma câmera e é definido pelo seu quadro inicial e final. O *shot* é considerado por Shanon et al. (1998) como o componente fundamental do filme. Uma cena é uma coleção de tomadas adjacentes focando sobre os mesmos objetos e descrevendo uma completa cadeia de ações usualmente no mesmo lugar e ao mesmo tempo. Por exemplo, uma pessoa andando pela rua e entrando na sua casa pode ser uma cena mesmo que mostrada por diferente ângulos ou câmeras, mas *shots* mostrando três pessoas diferentes andando na rua só pode ser considerado uma cena se o objeto principal dessa cena for à rua e não as pessoas. Normalmente essa cena é marcada em seu inicio e em seu final por uma transição ou corte.

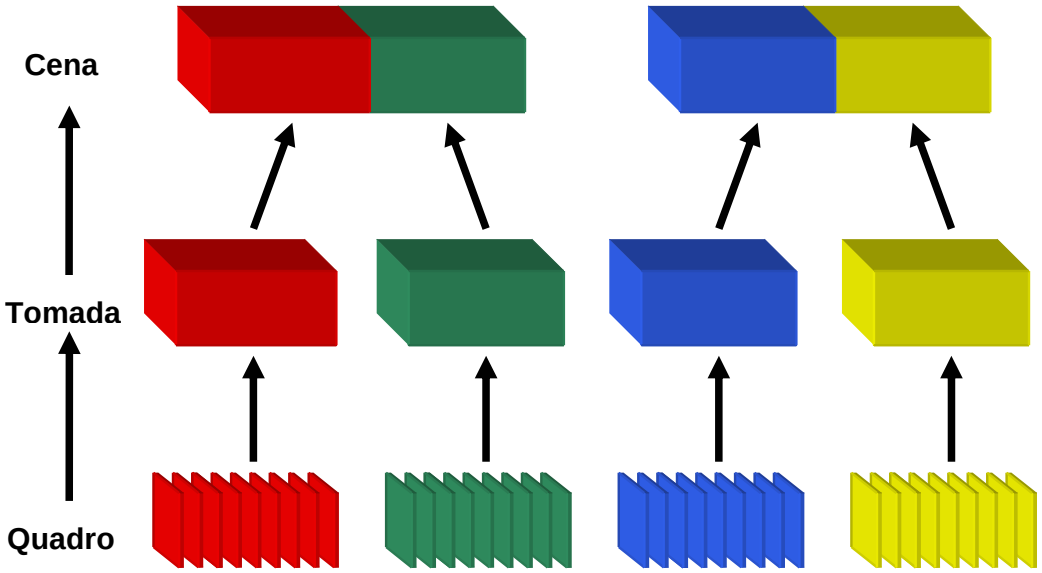


Figura 2.4- Conteúdo Estrutural Implícito dos Vídeos

2.1.3 Formatos de Vídeos

Os vídeos digitais em seu formato não compactado tornam-se arquivos muito grandes, uma saída para isso é comprimir para que ocupem menos espaço nos meios de armazenamento. Mas os computadores podem ter problemas em mostrar tais quadros caso estejam comprimidos, então se torna necessário que eles sejam descomprimidos para que se possa visualizá-los.

Comprimir vídeos através de *CODECs* que fazem o trabalho de codificar e decodificar os quadros, por um lado reduzem a necessidade em termos de armazenagem e facilitam que tais vídeos sejam disponíveis através de redes ou da Internet, mas aumentam a necessidade de processamento para reverter o processo de codificação aumentando a carga sobre o processador e quanto maior a compressão de dados, conseqüentemente maior será a perda de qualidade da imagem e do som.

Quanto aos formatos de vídeo podemos citar entre os mais comuns nos tempos de hoje segundo Ottewill et al. (1997):

- ***AVI:*** Este formato foi definido pela Microsoft e é o formato mais comum para dados de áudio/vídeo para computador.

- ***MPEG:*** MPEG também é conhecido como formato VCD, capaz de uma alta taxa de compressão e ainda assim consegue manter alta qualidade de imagem, mas tal qualidade ainda assim se mostra um pouco inferior a das fitas VHS.

- ***MPEG-2:*** MPEG-2 é o formato utilizado no DVDs. Este formato é uma nova versão do MPEG só que mais flexível e capaz de produzir vídeos de qualidade superior vídeos os quais podem ser reproduzidos em um simples computador (com 350Mhz ou mais).

- ***MPEG-4:*** MPEG-4 é um padrão para comprimir áudio e vídeo digital, os usos mais comuns deste formato é na transmissão de vídeos pela Internet, distribuição de vídeos em CDs e para conversação (videofone). O formato MPEG-4 absorve as características do MPEG-1 e do MPEG-2 e inclui suporte a composição orientada a objetos e gerenciadores de direitos digitais.

- ***MPEG-7:*** é um padrão para comprimir áudio e vídeo digital, porem não segue o esquema de codificação de vídeo e áudio como MPEG-1, MPEG-2 e MPEG-4. Este formato usa o XML para armazenar meta dados que podem ser anexados à linha de tempo do vídeo para marcar determinados eventos ou sincronizar letras a uma musica, por exemplo.

- ***MPEG-21:*** Este formato define uma “Linguagem de Expressão de Direitos” que gerencia o compartilhamento do conteúdo digital controlando os direitos/permissões/restrições entre o criador e o consumidor do conteúdo.

2.1.4 Framerate

O *framerate* (taxa de quadros) determina quantas imagens singulares são mostradas no intervalo de cada segundo de vídeo. O *framerate* é medido em *FPS,* quanto maior o *framerate* mais suave será a aparência do movimento durante a reprodução do vídeo. Mas as desvantagens de se usar mais quadros por segundo são que haverá uma necessidade maior de memória tanto para o armazenamento como para reprodução do vídeo.

2.1.5 Framesize

O *framesize* (tamanho do quadro) determina o tamanho que o vídeo será apresentado na tela, quanto maior a resolução mais detalhes haverá no vídeo. Esta medida representa a resolução do vídeo, descrevendo quantos *pixels* o filme tem na horizontal e na vertical. Mas em contrapartida novamente deparamos com o problema: quanto maior o tamanho do quadro mais memória será necessária para se armazenar este e também para reproduzi-lo, já que quanto maior o quadro, mais custoso é para o computador para redesenhá-lo.

2.1.6 Bitrate

O *bitrate* se refere quantos bits por segundo serão usados para descrever a imagem em um vídeo compactado. Esta propriedade determina a qualidade das imagens exibidas, quanto maior a taxa de bits melhor a qualidade.

2.1.7 Padrões NTSC e PAL

Esses são os padrões mais comuns nos dias de hoje. O formato *NTSC* que é usado na América do Norte e vários paises asiáticos e o *PAL* usado na maioria dos paises da Europa e do pacífico sul.

A diferença entre esses dois formatos é: o *framesize* dos vídeos PAL é de 352 x 288 com um *framerate* de 25 *fps* enquanto o padrão NTSC tem seu *framesize* de 320 x 240 e 29,97 *fps*.

2.1.8 Padrões Comuns de Vídeo

Certas combinações de *bitrate*, *framesize* e *framerate*, descrevem alguns padrões. Para melhor descrever tais padrões temos na Tabela 2.1 uma visão destes padrões e suas características.

Tabela 1 – Características dos Padrões Comuns de Vídeo

	VCD	*SVCD*	*DVD*	*HDDVD HDTV (WMVHD)*	*AVI DivX XviD WMV*	*AVI DV*
Resolução NTSC/PAL	*352x240 352x288*	*480x480 480x576*	*720x480* 720x576**	*1440x1080* 1280x720**	*640x480**	*720x480 720x576*
Compressão de Vídeo	*MPEG1*	*MPEG2*	*MPEG2, MPEG1*	*MPEG2 (WMV-MPEG4)*	*MPEG4*	*DV*
Tamanho/min	*10 MB/min*	*10-20 MB/min*	*30-70 MB/min*	*~150MB/min (~60MB/min)*	*4-10 MB/min*	*216MB/min*
Uso de CPU	*Baixo*	*Alto*	*Muito Alto*	*Extramente Alto*	*Super Alto*	*Alto*
Qualidade	*Boa*	*Muito Boa***	*Excelente***	*Soberba***	*Muito Boa***	*Excelente*

~ Bitrate aproximado, permitindo maior ou menor bitrate.
** Resolução aproximada, permitindo maior ou menor resolução.*
*** Qualidade dependente do bitrate e da resolução do vídeo.*

2.2 SEGMENTAÇÃO DE VÍDEOS EM TOMADAS

Normalmente cenas são marcadas em seu início e em seu final por uma transição A tomada por sua vez é composta por quadros com conteúdo suave e contínuo.

Devido ao grande avanço da tecnologia relacionada à produção de vídeo, vários tipos de transições são usados para indicar a mudança de espaço, tempo ou dar realce a eventos importantes. O número de transições possíveis é gigantesco, programas bem conhecidos como *Adobe Premier* (Adobe, 2006) ou *Ulead Media Studio* (Ulead, 2006), disponibilizam mais de cem tipos de transições. Tais transições ainda podem ter certos parâmetros alterados aumentando bastante o número de transições

possíveis. Mas conforme mostra Lienhart et al. (1997), na prática 99% de todos as transições se resumem em uma das três categorias: *cortes*; *fades*; *dissolves*.

Para uma melhor ilustração usaremos neste capitulo a Figura 2.5 e a Figura 2.6 como sendo respectivamente quadro inicial e quadro final da tomada.

Figura 2.5 – Quadro Inicial

Figura 2.6 – Quadro Final

2.2.1 Cortes

As transições abruptas (cortes) são simples, eles ocorrem em um único quadro, quando a câmera cessa e reinicia uma nova filmagem conforme mostrado em exemplo na figura 2.7.

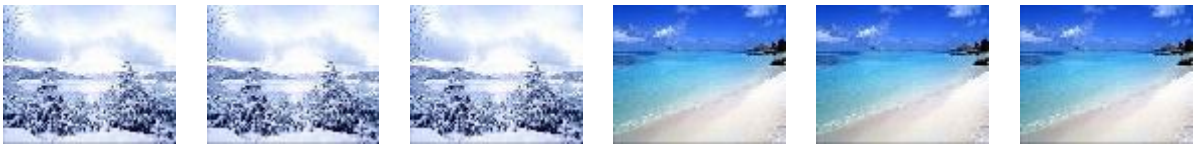


Figura 2.7 – "Corte" entre Cenas

2.2.2 Fades

As transições graduais mais comumente usadas são *fades.* O *fade out* consiste quando o quadro gradualmente é substituído por outro predominantemente preto enquanto o *fa*de in pode ser caracterizado pelo aparecimento gradual de um quadro a partir de um quadro de cor predominante preta. O *fade* pode apresentar algumas variações como, por exemplo, a cor predominante do quadro final no *fade out* ou o do quadro inicial no *fade i*n.

Durante o *fade*, uma imagem ou quadro de uma seqüência é substituído por um outro como visto na Figura 2.9. O fade que acontece entre duas cenas também é conhecido como *dissolve*. Um *fade in* ilustrado na Figura 2.10 pode ser caracterizado pela aparição gradual de uma imagem a partir de uma outra dominantemente preta e o *fade ou* em contrapartida ocorre quando a imagem gradualmente desaparece conforme visto na Figura 2.11, deixando uma imagem de cor dominantemente preta. Os *fades* podem apresentar variações de duração ou mesmo variações em relação à cor dominante da imagem que precede o *fade in* e resulta do *fade ou*t. *Fades* são usados freqüentemente para denotar transições de tempo e combinações destes podem indicar mudanças relativas entre tomadas. Os *fades* podem também ser usados para separar elementos diferentes em programas de televisão como o programa principal dos intervalos comerciais. (Lienhart et al., 1997)

Figura 2.8 – Exemplo de Dissolve

Figura 2.9 – "Fade In"

Figura 2.10 – "Fade Out"

2.3 SELEÇÃO DE QUADROS CHAVES

Um quadro chave ou *keyframe* é uma forma simples, porém eficiente maneira de se representar uma seqüência de vídeo. Vários autores recorrem a métodos baseados em *clustering*, os quais muitas vezes precisam de restrições de tempo e a seleção de apenas um quadro chave por *cluster*. Outra forma é fazer uma busca eficiente por uma quantidade especifica de quadros chaves em um vídeo.

Os quadros chaves costumam serem usados para representar partições homogêneas de vídeo. Para isto um quadro chave deve ser o mais similar possível aos outros quadros da partição ao qual pertence.

Keyframe é um frame que pode ser representado por ele próprio enquanto um não-*keyframe* ou um *delta-frame* são apenas representações das mudanças em relação ao *keyframe* e não podem ser representados sem este.

Os quadros que preenchem este espaço entre os *keyframes* são ditos como *inbetween* ou *delta-frames*, e constituem a maior parte dos quadros em um vídeo, esses quadros possuem pouca diferença entre si e em relação ao *keyframe*.

Uma tomada de vídeo com aproximadamente um segundo de duração, possui cerca de 30 imagens conforme mostra a Figura 2.12, já que usualmente os vídeos se baseiam em uma taxa de 30 quadros por segundo para proporcionarem uma impressão de movimento.

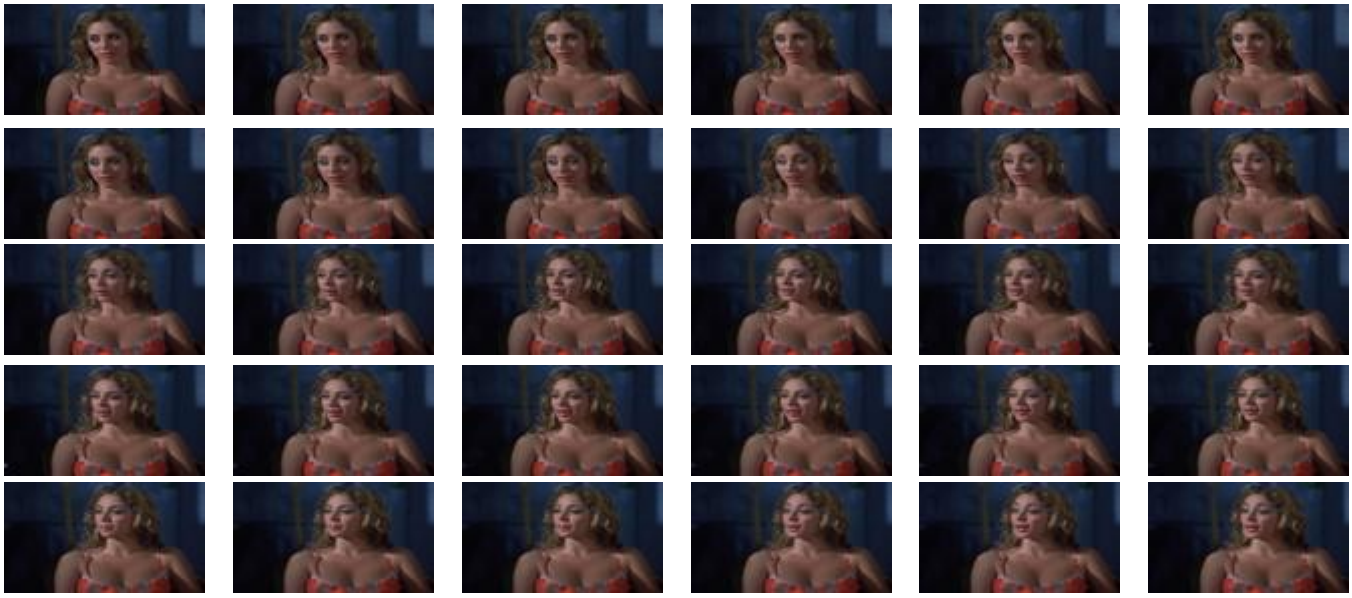


Figura 2.11 – Quadros Referentes a Um Segundos de Vídeo

Através da seleção de quadros chaves, podemos reduzir para um a quantidade muito menor o número de quadros a serem analisados, como podemos observar no exemplo da figura 2.13 que demonstra o resultado da seleção de quadros chaves para a seqüência representada na Figura 212.

Figura 2.12 – Keyframe detectado para seqüência mostrada na Figura 3.10

Devido a tal propriedade podemos usar estes quadros chave para representar uma seqüência de outros, os quais são dependentes da existência do *keyframe*.

2.4 CARACTERISTICAS DE COR

As cores por serem uma característica intrínseca das imagens, possuem um papel muito importante na recuperação desta. Para aumentar a eficiência no processamento, usualmente as cores da imagem são re-quantizadas com intuito de se diminuir o número total de cores presentes e facilitar o uso de histogramas. (Bueno et al., 2002)

O uso de histogramas de cores não acontece por acaso, há vários fatores que favorecem o uso destes, mas dentre esses fatores, há três que podemos ressaltar Pass et al. (1996):

- é computacionalmente simples e barata a resolução de imagens em histogramas de cores.
- pequenas alterações de movimentação de imagem pouco afetam os histogramas.
- objetos distintos freqüentemente possuem histogramas diferentes.

Dessa forma é natural que os histogramas de cores venham sendo estudados e implementados em sistemas de recuperação de imagens baseados em conteúdo, tanto acadêmicos (Hafner et al., 1995) quanto comerciais, como o QBIC (IBM, 2004).

2.4.1 Modelos de Cor

Newton desenvolveu um circulo de cores que definia um sistema de cores aditivas onde a cor branca era resultado de todas as cores do espectro visível, segundo ele haveriam nos olhos três receptores diferentes correspondendo cada um respectivamente pelas cores vermelho, verde e azul. Mais tarde Maxwell sugeriu um modelo de colores baseado na cromaticidade (matiz e saturação) que era invariável ao brilho.(Gattas, 2005)

Na análise de imagens, o processamento da cor é de grande importância para identificar e extrair características, devido estar presente em tudo o que é visível ao olho humano.

A cor está presente em tudo o que observamos. Este é um elemento essencial e desempenha funções múltiplas na visualização de imagens e cenas. A percepção da cor pelo homem é caracterizada pela interação da luz com seu sistema de visão. Desta forma, este interpreta as cores de maneira particular, dando-lhes um significado que

depende de condições psicofísicas. Para padronizar a especificação das cores, foram criados sistemas para representação de cor.

2.4.1.1 Modelo RGB

Em 1931 a *CIE* padronizou as cores primarias quanto a seu comprimento de onda, a cor vermelha λ_R = 700*nm* a cor verde λ_G = 346:1*nm* e a cor azul λ_B = 435:8*nm* resultando na representação de cores RGB que além de servir de base para os monitores coloridos passou a ser padrão para o armazenamento de imagens. Cada *pixel* é colorido por um trio de valores $(R;G;B) \in [0,1]$, daqui o espaço de cores RGB toma a forma de um cubo que é visto na Figura 2.13

Imagens no modelo RGB consistem em três planos de imagens independentes, um para cada cor primária, tais como: vermelho, verde e azul. Essas três imagens combinam-se sobre a tela, quando visualizadas em um monitor RGB e produzem uma imagem de cores compostas.

O modelo RGB está associado às superfícies emissoras de luz. Por isso, é um modelo quase universal empregado pelos equipamentos que manipulam a emissão de luz, tais como os monitores e os televisores a cores.

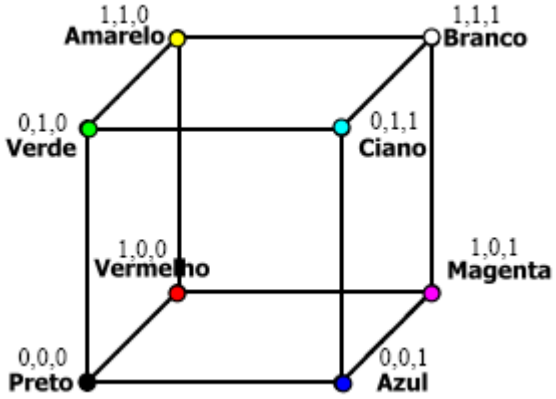


Figura 2.13 – Espaço RBG representado em Cubo

Os três parâmetros do modelo RGB definem um espaço tridimensional com direções ortogonais (R, G e B). Assim, está definido o espaço RGB. As cores deste espaço existem no subespaço em que $0 \le (R, G, B) \le 1$. Cada uma das cores primárias corresponde a um dos vértices do cubo localizados sobre os eixos do espaço, em que apenas uma das coordenadas não é nula. (Gonzalez et al., 2000)

2.4.1.2 Modelo HSV

O modelo HSV (*Hue*, *Saturation*, *Value*) de cores deve sua utilidade a dois fatores principais. Um é o componente de intensidade V que é desacoplado da informação de cor na imagem. O outro fator são os componentes: matiz e saturação que são relacionados à percepção humana de cores.

O termo *hue* distingue entre azul, verde, amarelo, vermelho; é a cor pura da imagem. A *saturation* da cor, por vezes denominada pureza ou saturação, indica o afastamento da cor. Uma cor vermelha ou azul puras são cores altamente saturadas, enquanto o rosa e as cores denominadas de pasteis são cores pouco saturadas. O

value é a intensidade da luz refletida pela superfície nos objetos, o brilho é a quantidade de luz emitida pelas superfícies de objetos luminosos. (Gonzalez et al., 2000)

Um exemplo gráfico do modelo de cor HSV pode ser visualizado na Figura 2.14.

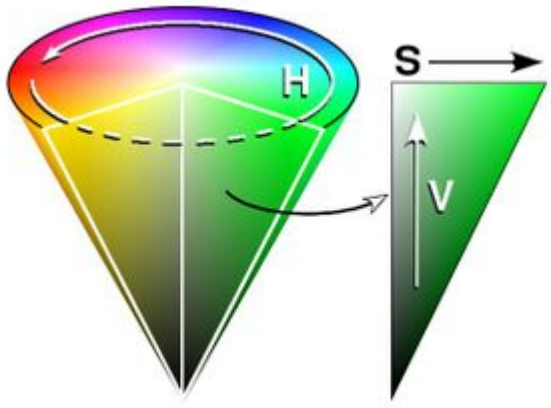


Figura 2.14 – Espaço de cor HSV

2.5 CARACTERISTICAS DE TEXTURA

A textura é considerada um padrão visual na qual há um grande número de elementos visíveis distribuídos de forma imparcial com variadas densidades. (Ramos et al., 2005)

A caracterização de textura varia com a intensidade em uma janela, tais como contraste, granularidade, direcionalidade e repetitividade. A análise da textura obtém os elementos presentes em uma imagem, determinando seu formato, e estimando as regras de posicionamento. Um exemplo de texturas está na Figura 2.15.

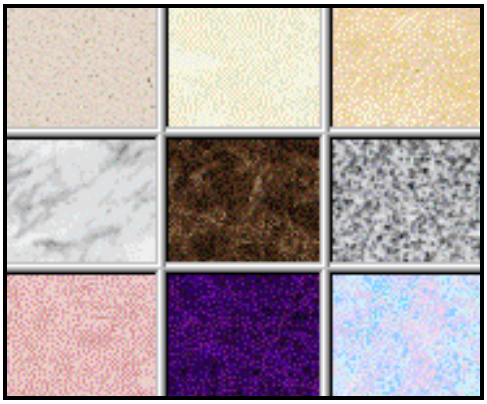


Figura 2.15 – Nove Diferentes Tipos de Textura.

Tratar texturas é diferente de se tratar cores, a textura não está diretamente relacionada a um *pixel* isolado, mas sim definidas sobre janelas ou regiões. A segmentação de uma imagem utilizando textura determina quais regiões da imagem possuem textura uniforme (Bueno et al., 2002).

Segundo Long et al. (2000), os modelos computacionais de característica de textura podem ser agrupados em três categorias:

- Estrutural: Os modelos estruturais caracterizam as texturas de acordo com o relacionamento local entre *pixels* de imagens.
- Estatístico: Modelos estatísticos categorizam texturas de acordo com medidas estatísticas de característica visual, tais como, grossura, granularidade, regularidade, entre outros.
- Espectral: Modelos espectrais caracterizam textura como propriedades de transformadas de Fourier ou nos resultados de filtragem das texturas por filtros apropriados.

2.6 CARACTERISTICAS DE MOVIMENTO

O Fluxo óptico pode ser definido como distribuição 2D da velocidade do movimento nos padrões de intensidade no plano da imagem conforme Horn (1986).

O campo do fluxo óptico consiste em um campo denso de velocidade onde a cada *pixel* no plano da imagem está associado um único vetor de velocidade. Para fins de visualização, o campo pode ser amostrado em uma malha e chamado de diagrama de agulhas (*needle map*) conforme mostra a Figura 2.17.

Figura 2.16 - Exemplo de Diagrama de Agulhas

Se for conhecido o intervalo de tempo entre duas imagens consecutivas, os vetores da velocidade podem ser convertidos em vetores de deslocamento e vice-versa (Ueda et al., 1991).

A análise do movimento é chamada de análise dinâmica de imagens e é baseada em um pequeno número de imagens em uma seqüência, algumas vezes duas ou três imagens. Esse caso é similar a uma análise estática de imagens e o movimento é atualizado através da correspondência entre pares de pontos de interesse nas imagens subseqüentes.

O fluxo óptico nem sempre corresponde ao verdadeiro campo de movimento por causa das mudanças de iluminação, mas representa uma aproximação dele. A

hipótese inicial na medida do movimento é que as estruturas de intensidade das regiões não sofrem grandes mudanças devido ao movimento em uma curta duração de tempo. Formalmente, se $I(x,t)$ é a função de intensidade da imagem, então:

$$I(x,t) \approx I(x+\delta x, t+\delta t) \quad (1)$$

Onde δx é o deslocamento de um local da imagem ou região em (x,t) após o tempo δt. Expandindo o lado esquerdo da Equação 2 na série de Taylor temos:

$$I(x,t) = I(x,t) + \nabla I \cdot \delta x + \delta t I_t + O^2 \quad (2)$$

Tendo $\nabla I = I_x + I_y$ e I_t como a derivada parcial de primeira ordem de $I(x,t)$ e O^2, o termo de segunda ordem que pode ser negligenciado, assim subtraindo $I(x,t)$ nos dois lados da Equação 2, ignorando O^2 e dividindo por δt obtemos:

$$\nabla I \cdot v + I_t = 0 \quad (3)$$

Na Equação 3 temos $\nabla I = I_x + I_y$ como o gradiente de intensidade espacial e $v = (u,v)$ representando a velocidade da imagem. A Equação 3 é conhecida como "*Equação de Restrição do Fluxo Ótico*" definindo uma restrição local no movimento da imagem. Conforme pode ser observado na Figura 2.18, $v_\perp$ é definido como um vetor perpendicular a linha de restrição.

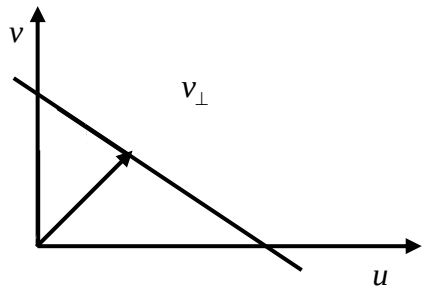


Figura 2.17 - Equação de Restrição do Movimento.

O fluxo óptico pode ser exatamente extraído do movimento da imagem, se as seguintes condições estiverem satisfeitas: iluminação uniforme, refletância Lambertiana (brilho igual em todas as direções) e translação paralela ao plano da imagem (Barron et al., 1994).

2.7 FUNÇÕES DE SIMILARIDADE

Funções de similaridade são os métodos usados para avaliar a similaridade ou dissimilaridade entre imagens, podendo ser distancia euclidiana, desvio padrão, intersecção de histogramas entre outros.

Existem muitas classes e varias classes de características e, portanto varias técnicas para medi-las, formadas pela combinação ou distribuição de medidas simples como, por exemplo:

- ***Tamanho do Objeto:*** Área, volume, perímetro, superfície que podem ser obtidos pela contagem de pixels.

- ***Forma do Objeto:*** A forma pode ser obtida pela caracterização da borda do objeto através de descritores de Fourier, momentos invariantes medidas de forma ou esqueletos.

- ***Cor do Objeto:*** Descritores no espaço de cor, densidade ótica integrada, cores absolutas ou relativas podem ser usadas para obter a cor de um objeto.

- ***Aparência/Textura do Objeto:*** Variações de cor em pixels vizinhos, matrizes de co-ocorrência, medidas fractais, características estatísticas geométricas são uma opção para medida de aparência e/ou textura.

- ***Parâmetros Distribucionais:*** Momentos, média, mediana, variância que podem ser descritos a partir de distribuições estatísticas a partir de um ou mais características.

2.8 REVOCAÇÃO E PRECISÃO

Na implementação final de um sistema de recuperação de informação por conteúdo, é necessário fazer uma avaliação do sistema. Estes sistemas necessitam da avaliação de quanto preciso é o conjunto de resposta. Assim, este tipo de avaliação é denominado Avaliação do Desempenho da Efetividade.

A medida chamada de Efetividade foi proposta por Smeulders et al. (2000) para avaliar os métodos de recuperação de imagem.

Para avaliar o desempenho do algoritmo de recuperação de informação, a eficiência é medida em termos de revocação e precisão. Para entender melhor o significado dessas medidas:

Estes sistemas classificam os documentos recuperados para cada consulta, de acordo com uma ordem de relevância gerando um vetor resultado. Avalia-se então através da comparação das respostas geradas por este sistema e o conjunto ideal de respostas. Para isso, o vetor resultado é examinado e comparado com o conjunto ideal, obtendo-se dois índices de avaliação: precisão e revocação.

A precisão é a fração dos documentos já examinados que são relevantes, e revocação é a fração dos documentos relevantes observada dentre os documentos examinados.

Seja I o conjunto de resposta ideal e R o conjunto do resultado recuperado pelo sistema de recuperação de vídeos por conteúdo. Tendo também $|I|$ e $|R|$ como sendo o número de elementos do conjunto I e R respectivamente. Desta forma temos:

$$\text{Re}\,vocação = \frac{I \cap R}{|I|} \qquad (4)$$

$$\text{Pr}\,ecisão = \frac{N \cap R}{|R|} \qquad (5)$$

3 ESTADO DA ARTE

Neste capítulo, são abordadas técnicas presentes na literatura e relacionadas à proposta de indexação de vídeos apresentada neste trabalho.

3.1 DETECÇÃO DE TOMADAS

A detecção de transições em vídeos é fundamental para praticamente todo tipo aplicação de análise de vídeo por possibilitar a segmentação do vídeo em seus componentes básicos: as tomadas.

Sendo assim a seguir são abordados os métodos mais populares para a detecção de tomadas

3.1.1 Diferenças de Pixels

A forma mais fácil de detectar se dois quadros são significativamente diferentes é contando o número de *pixels* que mudam acima de um limiar. Este total pe comparado com um segundo limiar para determinar se uma transição é encontrada. Mas este método é sensível ao movimento de câmera.

Zhang et al., (1993) apresenta-se um método que implementa um passo adicional onde um filtro da mediana de 3x3 é usado para atenuar os efeitos do movimento e ruído. Apesar disto, o método é lento e nota-se também que o valor dos limiares deve ser ajustado manualmente tornando o processo muito pouco prático.

Shahraray, (1995) utiliza o mesmo processo inicial mas dividiu as imagens em doze regiões. Para cada região efetua uma procura pela vizinha de maior similaridade em outra imagem. A diferença de *pixels* para cada região é ordenada e as diferenças de regiões provêem a medida de diferença entre as imagens. Algumas transições graduais podem ser detectadas pela medida da diferença acumulativa dos valores de imagens consecutivas.

Boreczky et al., (1996) computaram o que eles chamaram de imagens cromáticas (*Chromatic Images*) pela divisão da variação no nível de cinza de cada *pixel* entre duas imagens pelo nível de cinza do *pixel* da segunda imagem. Durante os *dissolves* e *fades*, esta imagem cromática assume um valor razoavelmente constante. Infelizmente esta técnica é muito sensível ao movimento de câmera quanto ao dos objetos presentes no quadro.

3.1.2 Diferenças Estatísticas

Métodos estatísticos expandem a idéia da diferença de *pixels* dividindo a imagem em regiões e comparando os *pixels* nestas regiões por meio de medidas estatísticas.

Kasturi et al. (1991) computam uma medida baseada na media e desvio padrão dos níveis de cinza nas regiões da imagem. Este método é razoavelmente tolerante a ruído, porém é lento devido a complexidade das formulas estatísticas envolvidas além de detectar muitas vezes elementos a na verdade não são transições.

3.1.3 Histogramas

Histograma é o método mais comum usado para se detectar transições. O mais simples método baseado em histogramas calcula o nível de cinza o histograma de cores de duas imagens. Se a diferença entre os dois histogramas estiver acima de um limiar assume-se que ali há uma transição. Ueda et al., (1991) demonstram como usar a mudança de valores nos histogramas para encontrar as transições.

Nagasaka et al. (1992) comparam várias medidas estatísticas simples baseadas no histograma do nível de cinza e de cores da imagem. Os melhores resultados são obtidos particionando as imagens em dezesseis regiões e abandonando as oito maiores diferenças encontradas a fim de reduzir os efeitos indesejados causados pelo movimento e pelo ruído. Já em (Swanberg et al., 1993) usam as diferenças dos histogramas das regiões ponderado de acordo com a região da imagem. Este método funciona bem em caso de vídeos em que há uma estrutura espacial regular.

Zhang et al. (1993) utilizam uma técnica que combina a comparação da diferenças de *pixels* medidas estatísticas e vários métodos baseados em histogramas mantendo assim um balanço entre precisão e velocidade, mas para detectar certos tipos de transições graduais como *wipes* e *dissolves* é necessário usar dois diferentes limiares, tais limiares representam os valores permitidos para o inicio da seqüência de transição gradual e seu final.

3.1.4 Diferenças de Compressão

Little et al. (1993) usam as diferenças no tamanho dos quadros JPEG comprimidos para detectar transições como suplemento à anotação manual. Enquanto isso Arman et al. (1994) usam a comparação de um pequeno número de regiões conectadas bem como diferenças nos coeficientes da transformada discreta dos co-senos (DCT) nos quadros compactados como medida de similaridade.

3.1.5 Perseguição de Cantos

Zabih et al., (1995) alinham quadros consecutivos para reduzir os efeitos de movimentação da câmera. Feito isso detectam os cantos presentes no quadro e comparam o número de cantos e suas posições. O percentual de cantos que entram e saem entre dois quadros é computado e as transições são encontradas quando grandes percentuais são atingidos, este método é bastante robusto na detecção de cortes que os histogramas além de ser bem menos sensível ao movimento.

3.2 CARACTERISTICAS DE TEXTURA

Os modelos estatístico e espectral são freqüentemente usados em sistemas de recuperação de imagem baseados por textura.

Oliveira et al. (2002) propõem o armazenamento de informações de regiões com a mesma cor na imagem (regiões cromáticas), como o tamanho, posição e limites com outras regiões.

Já Stricker et al. (1996) usam histogramas para codificar as cores semelhantes, e computar a informação geométrica em indexação da imagem colorida. Esse método produz um espaço de cores distintas de 256 elementos, mas, não é robusto o suficiente para imagens relacionadas com textura.

Long et al. (2000) abordam o uso texturas com escalas, orientações, intensidade e contraste. Para mostrar a aplicação desse espaço perceptual, características da textura são extraídas usando filtros de Gabor e uma rede neuronal *feedforward multi-layer* é treinada para mapear as características de Gabor nesse espaço perceptual.

3.3 EXTRAÇÃO DE CARACTERÍSTICAS DINÂMICAS

A análise do movimento é chamada de análise dinâmica de imagens e é baseada em um pequeno número de imagens em uma seqüência, algumas vezes duas ou três imagens. Esse caso é similar a uma análise estática de imagens e o movimento é atualizado através da correspondência entre pares de pontos de interesse nas imagens subseqüentes.

O fluxo óptico nem sempre corresponde ao verdadeiro campo de movimento por causa das mudanças de iluminação, mas representa uma aproximação dele. A hipótese inicial na medida do movimento é que as estruturas de intensidade das regiões não sofrem grandes mudanças devido ao movimento em uma curta duração de tempo.

Os métodos a seguir explanam sobre as diferentes abordagens da extração do fluxo ótico.

3.3.1 Métodos Diferenciais

As técnicas diferenciais computam a velocidade da imagem a partir da derivada espaço-temporal das intensidades da imagem. Beauchenmin et al. (1995) descrevem que o domínio da imagem é assumido conseqüentemente sendo continuo (ou diferençável) no espaço e tempo. Os métodos globais e locais de primeira e segunda ordem se baseiam na Equação 2.3. Métodos globais usam restrições globais adicionais, usualmente um termo de regularização e suavização para calcular fluxo ótico denso em regiões grandes ou imagens inteiras. Métodos locais usam a informação da velocidade normal em vizinhos locais para efetuar a minimização e encontrar o melhor valor para v. A precisão das técnicas diferenciais depende principalmente da estimação das derivadas parciais da função de intensidade. Apesar do método de diferenças finitas ser simples ele não consegue fazer a distinção entre dados verdadeiros e ruídos e para eliminar ou reduzir esses problemas é realizada uma pré-suavização da imagem com um filtro gaussiano, por exemplo.

3.3.2 Métodos Baseados em Freqüência

Beauchenmin et al. (1995) trazem que neste método de cálculo do fluxo ótico usa-se filtros sensíveis à orientação no domínio de Fourier em imagens com variação de tempo. Entre as vantagens trazidas por este método podemos citar a capacidade de efetuar o cálculo do fluxo ótico em situações onde as abordagens por *matching* seriam incapazes de fazê-lo. Por exemplo, calcular o movimento a partir de padrões aleatórios de pontos pode ser difícil de se efetuar a partir de técnicas baseadas em características ou baseadas em correlação, mas no espaço de Fourier, o resultado da energia da orientação é mais eficiente para isso.

3.3.3 Métodos Baseados Correspondência

A extração de vetores de velocidade baseada em correspondência de regiões determina o movimento correto de uma janela de *pixels* simulando o movimento da janela para cada posição (x, y) e considerando qual a maior similaridade entre os valores das janelas nos quadros no instante t e $t+1$.

Barron et al. (1994) demonstram que, Se uma função que retorna um valor proporcional à similaridade das duas janelas em dois quadros diferentes (dada, por exemplo, pela como a Soma das Diferenças Absolutas dos valores de intensidade dos *pixels* - SAD), então a correspondência $M(x, y, s, w)$ da janela no ponto (x, y) e todas aquelas nos deslocamentos (s, w) são calculadas. Dentre todos os deslocamentos

possíveis, aquele que minimizar o critério da função irá atribuir a sua direção e módulo ao vetor de velocidade do pixel (x, y).

Este é um algoritmo bastante robusto já que não requer que os valores de intensidade dos *pixels* entre os quadros tenham valores próximos. Por exemplo, se houver uma mudança de iluminação certamente o valor de mudaria para todos os valores de (s, w) mas isso não afetaria o resultado da melhor combinação.

3.4 SISTEMAS DE RECUPERAÇÃO DE VÍDEO POR CONTEÚDO

O "*Video Retrieval and Sequencing System*" segue a metodologia proposta por Chua et al. (2001), é capaz de suportar todo o processo de recuperação de vídeo, bem como a segmentação da seqüência de vídeo em tomadas. O movimento também é avaliado, mas somente para obter-se informações espaciais como posição da câmera tipo de movimento (*dolly*; *hand-held*; *pan*; *tilt*; *zoom-in*; e *zoom-out*). Portanto a necessidade deste método está na necessidade do uso da informação textual que precisa ser inserida manualmente para descrever a tomada e que sua recuperação não pode ser feita a partir de uma amostra, mas sim de uma *query* textual.

O *ImageMiner* baseia-se no método de Alshuth et al. (1998), nele os vídeos são disponibilizados em dois passos. Inicialmente as tomadas são extraídas do vídeo usando-se um método baseado em histograma, em seguida, usa uma técnica de mosaico para gerar uma imagem para cada tomada detectada na seqüência de vídeo. Essas imagens irão conter todas as informações necessárias para serem analisadas pelo sistema que gera descrições de contexto baseadas na cor, na textura e nos contornos.

A proposta de Cascia et al., (1996) baseia-se na busca em cor e textura. A cor é representada num histograma no espaço *RGB*, as características de textura usadas são extraídas da matriz de co-ocorrência do nível de cinza da imagem sendo elas a probabilidade máxima e a uniformidade. Como critério de comparação entre os quadros utiliza-se a distância entre os histogramas. A medida de distância usada para texturas não foi mencionada. Este processo pode ser aplicado a um determinado quadro de vídeo ou uma imagem, o método retorna os quadros ou imagens por ordem de similaridade.

Já o *ShoeBox* de Mills et al. (2000) faz a anotação de palavras chaves por reconhecimento de voz. O sistema também faz uma partição da imagem sobre as quais determina a cor média no espaço *HSV* e calcula a variação em cada um dos canais de cada região. Assim este sistema pode fazer procura por anotações faladas ou por regiões semelhantes, no caso das regiões, usa-se o cálculo da distância euclidiana entre o vetor de características das duas regiões da imagem.

O *WebSeek* proposto por Smith et al. (1995) faz buscas baseadas em texto ou cor em catálogos de imagens e vídeos. A cor é representada por um histograma normalizado no espaço de cor *HSV*. Para tal, utiliza-se os mesmos padrões do *ViualSeek* que decompõe cada imagem em regiões de cores dominantes. Para cada região, as características e propriedades espaciais são guardadas para consultas. Uma imagem é considerada mais similar à outra quanto mais arranjos de regiões similares ela tiver. No *WebSeek*, o usuário inicia uma consulta escolhendo a partir dos títulos disponíveis num catalogo, a distância entre o histograma consultado e o histograma alvo. Outro item interessante é a possibilidade do usuário poder qualificar como positivos ou negativos os exemplos retornados em uma busca, re-balanceando a relevância dos resultados através do feedback do usuário.

A velocidade e facilidade que a seleção de apenas alguns quadros dentro do vídeo traz para a recuperação de informação são alguns atrativos que justificam a grande de soluções que se tem valido deste artifício.

Conforme é possível observar na literatura, ainda é pequena a quantidade de métodos que levem em consideração aspectos dinâmicos, a maior parte utiliza informações estáticas e descrições textuais na hora de analisar os quadros selecionados.

4 METODOLOGIA

Devido a esta grande quantidade de informação, usa-se uma metodologia de seleção de quadros chaves para evitar que se faça desnecessariamente a extração de características em todos quadros do vídeo. Esta seleção de poucos quadros ao invés de se usar todos eles, serve como processo de seleção de dados. Esta seleção é capaz de evitar o excesso de informações a armazenar após a extração de características e conseqüentemente a processar durante a recuperação.

Com poucos quadros a serem analisados, usa-se extração das características estáticas que pode ser feita de várias formas, mas neste trabalho a extração de características se divide em dois tipos de características: estáticas e dinâmicas. As características estáticas que são extraídas focando-se em cor e textura através de wavelets no espaço de cor *HSV*. Já as características dinâmicas são representadas pela estimativa do fluxo ótico do quadro chave em relação aos quadros próximos a este na tomada. Para uma melhor representação dos atributos de movimento neste trabalho são feitos experimentos para determinar a melhor forma de extrair as características do fluxo ótico. Também neste trabalho ambos atributos são devidamente ponderados para que seja possível determinar a influência destes atributos na recuperação e encontrar a combinação que traz melhores resultados para a recuperação de vídeos por conteúdo.

De posse destas informações são criadas assinaturas para cada chave contendo as características dinâmicas e estáticas. Tais características são usadas durante o processo de recuperação onde as assinaturas são comparadas e recebem um *score* de acordo com o seu grau de similaridade para com o quadro de referência para por final o método trazer ao usuário os resultados com maiores *scores*.

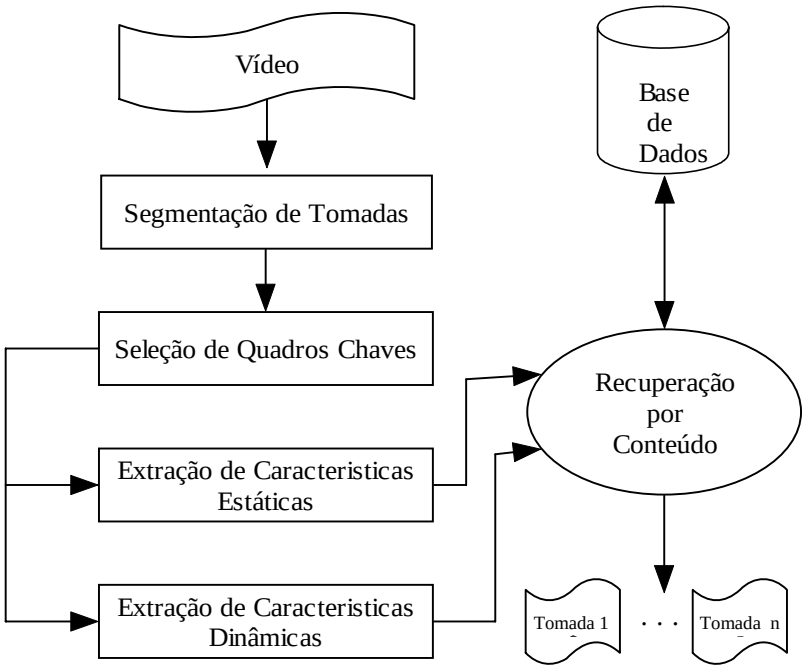


Figura 4.1 - Estrutura da Metodologia Proposta

4.1 SEGMENTAÇÃO DE TOMADAS

A detecção de transições em vídeos é fundamental para praticamente todo tipo aplicação de análise de vídeo por possibilitar a segmentação do vídeo em seus componentes básicos: as tomadas.

Como as tomadas são limitadas por transições, para se definir uma tomada, basta que se encontre tais transições. Tais transições conforme mostrado anteriormente possuem características peculiares sendo assim necessitam técnicas diferentes para sua detecção. A tabela a seguir mostra que tipo de transição é detectado por qual técnica.

Tabela 2 - Transições e Métodos de Detecção (Lienhart, 1999)

	Cortes	Fades	Dissolve
*Diferença de Cor dos His*togramas	*x*		
Taxa de Mudança de Contornos	*x*	*x*	*x*
Desvio Padrão da Intensidade dos Pixels		*x*	
Contraste			*x*

Nota-se que a metodologia de detecção de transições baseada na "Taxa de Mudança de Contorno" se mostra mais genérica, sendo capaz de detectar os três tipos de transições sendo por tanto o método adotado.

A Taxa de Mudança de Contorno (*ECR*) é definida da seguinte forma: Seja σ_n o número de *pixels* contorno no quadro n, X_n^{in} e X_{n-1}^{out} os *pixels* de contorno que presentes nos quadros n e $n-1$, respectivamente. Então:

$$ECR_n = \max\left(\frac{X_n^{in}}{\sigma_n}, \frac{X_{n-1}^{out}}{\sigma_{n-1}}\right) \qquad (6)$$

A Equação 6 resulta na taxa de mudança de contorno entre os quadros n e $n-1$, estas taxas podem variar entre 0 e 1. Os contornos são calculados por uma detector de contornos de Canny (1986).

De acordo com Zabih et al. (1995), cortes, *fades, dissolves* e *wipes* exibem características padrão na série temporal da *ECR.* Cortes são reconhecidos por picos isolados, durante *fade-ins* e *fade-outs* o número de contornos que entram e saem respectivamente é muito grande enquanto isso durante nas transições *dissolve* inicialmente os contornos que saem do primeiro quadro são predominantes e em seguida os contornos de um segundo quadro passam a predominar. Esses três casos podem ser vistos nas Figura 4.2, 4.3 e 4.4.

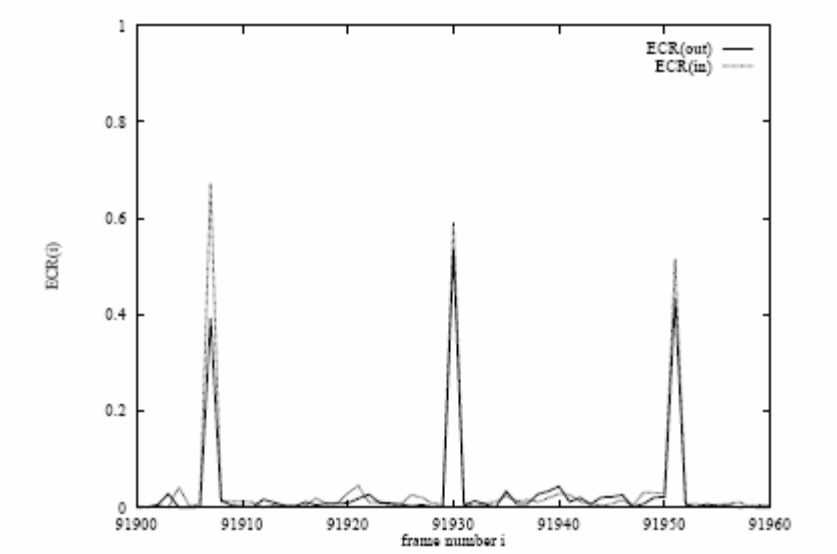


Figura 4.2 - Padrões Típicos do ECR em Cortes

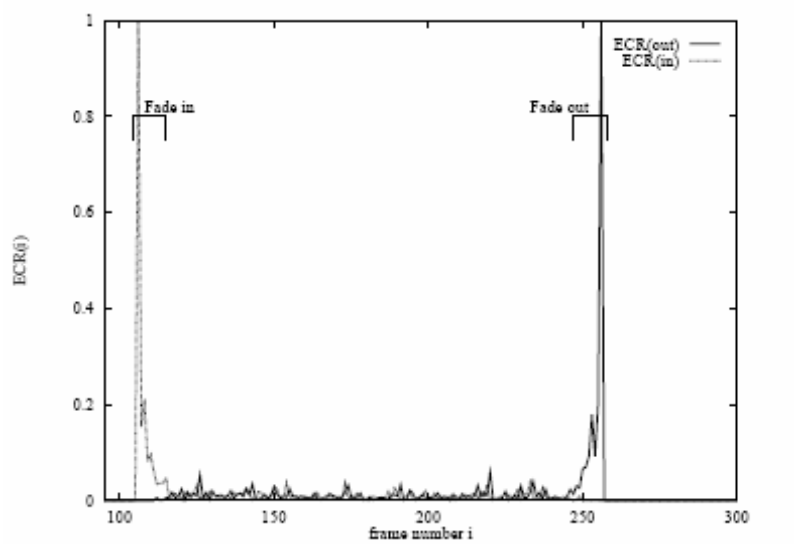


Figura 4.3 - Padrões Típicos do ECR em Fades

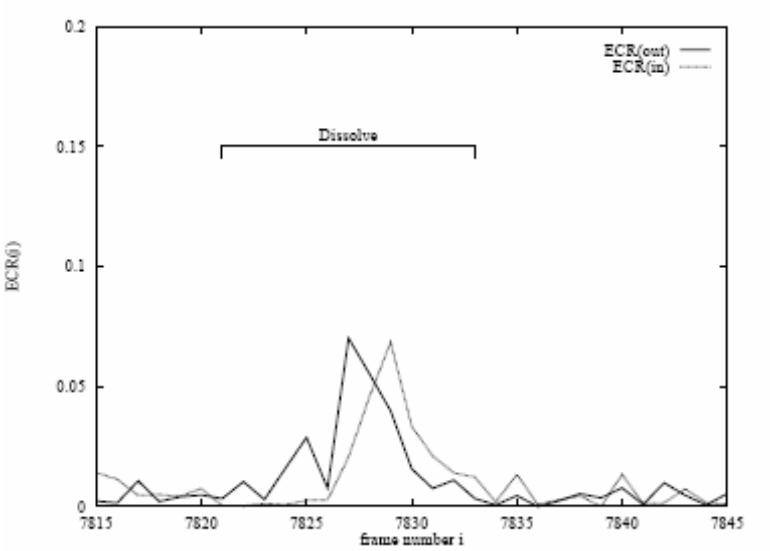


Figura 4.4 - Padrões Típicos do ECR em Dissolves

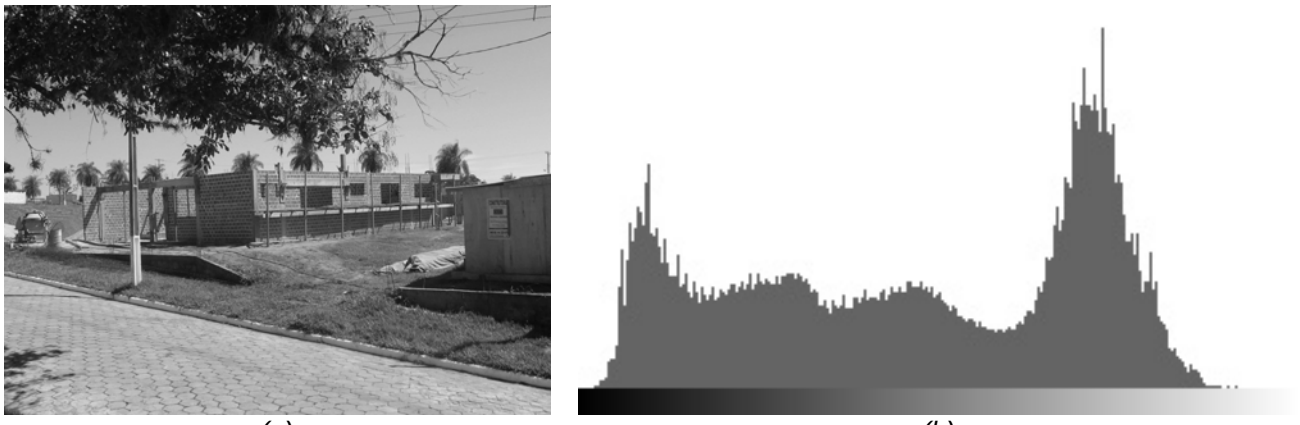
4.2 SELEÇÃO DE QUADROS CHAVES

Partindo do principio que os quadros presentes nos vídeos nada mais são que simples imagens, nós podemos tratar então cada quadro de um vídeo como uma imagem.

Uma imagem é descrita pela função *f(x,y)* com intensidade de luz sobre a cena, sendo seu valor, em qualquer ponto de coordenadas espaciais *(x,y)*, proporcional ao brilho da imagem naquele ponto. Em uma imagem cujas informações são apresentadas em intervalos ou bandas distintas de freqüência, é necessária uma função f*(x,y)* para cada banda. É o caso de imagens coloridas padrão RGB, que são formadas pela informação de cores primárias, como o vermelho (*Red*), verde (*Green*) e azul (*Blue*). E as imagens no formato HSV que são formadas pela informação do matiz (*Hue)*, saturação (*Saturation)* e intensidade de luz (*Value)*. Para o processamento de uma imagem é fundamental representar sua informação num formato adequado ao tratamento computacional. Uma imagem pode ser representada por uma matriz, em

que os índices de linha e coluna referenciam o brilho médio amostrado no ponto correspondente da cena (Facon, 1996).

A visualização da distribuição de dados de uma imagem é feita por um histograma que dá a distribuição das informações graficamente, veja exemplo na Figura 4.5.



(a) *(b)*

Figura 4.5 – Imagem original em Tons de Cinza (a)e seu Histograma (b)

O histograma de uma imagem corresponde a uma tabela que dá para cada nível de cinza, o número de *pixels* correspondentes na imagem. Quando o histograma fornece somente o número de *pixels* e não a localização desses, ele permite dar uma descrição global da imagem.

A seleção do quadro chave é feita a partir do método relatado por Rasheed et al. (2003). Neste método a seleção acontece em dois passos. Primeiro as tomadas já segmentadas são analisados quadro a quadro por meio de intersecção de histogramas os quadros aqui selecionados são dados como possíveis quadros chaves. Depois desta etapa concluída os prováveis quadros chaves são comparados entre si, de forma a reduzir a quantidade de quadros com informação redundante e gerar um conjunto de quadros chaves apenas com quadros distintos.

Rasheed et al. (2003) afirmam que para a detecção de um conjunto de quadros referentes a mesma câmera, pode ser usado um cálculo de intersecção de histogramas dos canais *HSV.* O histograma é composto por 16 *bins* (8 para o canal H, 4 para o canal S e 4 para o canal V).

Seja f_i a representação de um quadro em uma tomada e H_i o seu respectivo histograma e *b* a referencia aos canais *HSV.* Desta forma podemos definir a intersecção entre dois quadros consecutivos pela Equação 7.

$$D\left(f_i, f_{i+1}\right) = \sum_{b \in allbins} Min\left(H_i(b), H_{i+1}(b)\right) \qquad (7)$$

Um novo quadro chave é detectado em função de um limiar T_{color} que define quando se tem uma intersecção insuficiente entre os quadros i e seu quadro seguinte como mostra a Equação 8.

$$D\left(f_i, f_{i+1}\right) < T_{color} \qquad (8)$$

Os quadros que satisfazem esse limiar são incluídos no conjunto de prováveis quadros chaves que pode ser representado como $PK_i = \{f_1, f_2, \ldots, f_{n-1}, f_n\}$. Cada tomada pode ser representada por um conjunto de quadros chaves K_i, no qual todos os quadros devem ser distintos.

Para esta solução inicia-se o conjunto K_i como vazio e o quadro central do conjunto PK_i é adicionado a ele. Em seguida todos os quadros de PK_i serão comparados aos do conjunto K_i e caso a interseção entre o quadro de referência e todos os do conjunto K_i sejam menores que um limiar, o quadro de referencia é inserido no conjunto K_i conforme podemos observar na Equação 9.

$$K_i \leftarrow \left\{ f^{(n/2)} \right\}$$
$$\textbf{\textit{if}}\ \max\left(D\left(f^j, f^k\right)\right) < T_{color}\ \forall f^k \in K_i \qquad (9)$$
$$\textbf{\textit{then}}\ K_i \leftarrow K_i \cup \left\{ f^j \right\}$$

Ao término do processamento de todos os quadros da tomada, o conjunto K_i contém apenas os quadros chaves da tomada.

Um problema encontrado na seleção de quadros chaves para tomadas com grande movimentação é que, por exemplo, uma explosão acontece em vários quadros consecutivos e não somente em um deles, dificultando a escolha. Outras classes também sofrem com o mesmo problema, pois a seleção é feita no inicio da mudança brusca de conteúdo. Essa presença do conteúdo disperso em vários quadros e não condensado em apenas um fator que trás complexidade e inexatidão na escolha do quadro chave.

4.3 EXTRAÇÃO DE CARACTERÍSTICAS ESTÁTICAS

Devido ao fato de que as imagens podem possuir a mesma proporção de cores, mas diferentes distribuições espaciais. Conclui-se que inserir as informações espaciais serve como informação que auxilia o processo de recuperação nestes casos. Assim, uma maneira de fornecer informação espacial é dividir a imagem em sub-imagens, e indexar cada uma dessas partes (Stricker et al., 1996).

A Figura 4.6 mostra alguns exemplos de padrões tendo a mesma proporção de cor, mas diferente distribuição espacial.

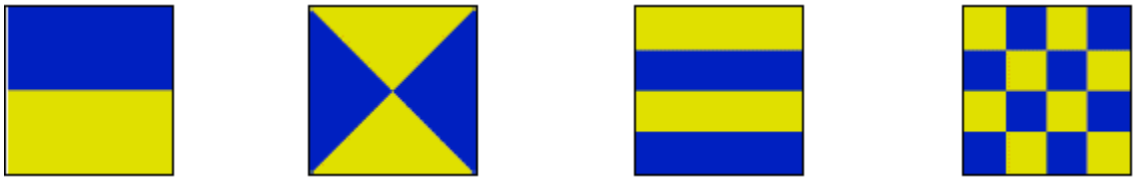


Figura 4.6 – Figuras mesma proporção de cor, mas diferente distribuição espacial.(Ramos et al., 2005).

A vantagem de se trabalhar com distribuição espacial é que ela fornece informações significativas através dos *pixels* de cada porção da imagem que for particionada. Neste trabalho está se usando a distribuição espacial fazendo uma divisão das imagens em nove partes iguais.

Optou-se por essa divisão pelo fato de poder localizar elementos que mesmo dispostos em regiões diferentes da imagem possam ser comparados e assim um *ranking* pode ser feito.

Como se propõe trabalhar com imagens quaisquer, a divisão em nove partes possibilita imagens com muitos tipos de elementos, e não só aquela com elemento central mais importante. (Ramos et al., 2005)

Assim é feita a divisão espacial do quadro chave em nove partes iguais, sendo três colunas por três linhas. Esta ação ajuda eliminar os problemas de distribuição espacial vistos mostrados anteriormente. Optou-se também por essa divisão pelo fato de poder localizar elementos que mesmo dispostos em regiões diferentes da imagem possam ser comparados e assim um *ranking* pode ser feito. Como se propõe trabalhar com quadros quaisquer, a divisão em nove partes possibilita quadros com muitos tipos de elementos, e não só com elemento central mais importante.

4.4 CONVERSÃO DE RGB PARA HSV

A vantagem dessa representação de cor está na possibilidade de separar os canais. Esse fato torna o modelo HSV uma ferramenta ideal para o desenvolvimento de algoritmos de processamento de imagens.

As cores nos modelos HSV são obtidas a partir das informações RGB com respeito aos valores normalizados do vermelho, verde e azul, dados por:

$$r = \frac{\mathrm{Re}\, d}{255} \tag{10}$$

$$g = \frac{Green}{255} \tag{11}$$

$$b = \frac{Blue}{255} \tag{12}$$

Após obter os canais R, G e B normalizados, as equações abaixo são utilizadas para o cálculo da intensidade, saturação e matiz.

$$V = \frac{(r+g+b)}{3} \tag{13}$$

$$S = 1 - \frac{3}{(r+g+b)} \times \min(r,g,b) \tag{14}$$

$$H = \arctan\left(\frac{\sqrt{(r-g)+(r-b)}}{\sqrt{(r-g)^2 + (r-g)(g-b)}} \right) \tag{15}$$

Assim temos a partir de agora, um quadro chave dividido em nove pedaços sendo que cada um dos pedaços é separado em três canais (H, S, V).

4.5 TRANSFORMADA WAVELET

A transformada *wavelet* emprega as funções *wavelet*, estas funções tem excelente localização espaço-temporal, permitindo desenvolver decomposições *wavelet* com grande variedade de funções básicas, e também permite enfatizar redundâncias ou eliminá-las através dos níveis de decomposição.

A decomposição de Mallat define a aplicação de operações de convolução do sinal com filtros *QMF's* (*Quadrature Mirror Filters*) e subamostragens (*downsamplings*). A Figura 4.7 ilustra o processo de decomposição.

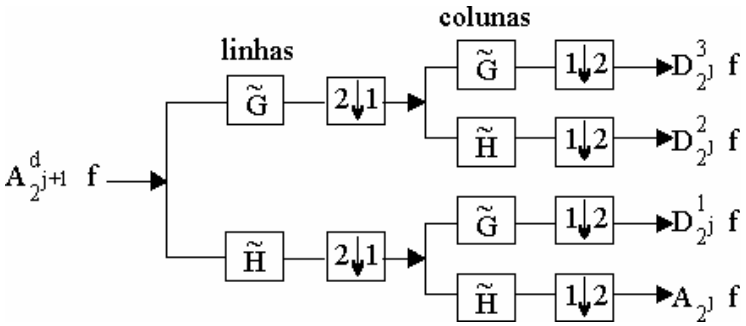


Figura 4.7 – Diagrama ilustrando o processo de decomposição

Esta decomposição proposta por Mallat define o uso de operações de convolução do sinal com filtros passa-baixa (H) e passa-alta (G) nos dados originais, seguidos de operações de subamostragem de linhas (1↓2) e colunas (2↓1), dependendo do caso. A operação (1↓2) mantém uma linha de duas e a (2↓1) mantém uma coluna de duas.

Desta forma o sinal de entrada $A^{d}_{2^{j+1}} f$ é decomposto em quatro conjuntos de coeficientes: $A^{d}_{2^j} f$, $D^{1}_{2^j} f$, $D^{2}_{2^j} f$ e $D^{3}_{2^j} f$, onde as $D^{i}_{2^j} f$ representam as bandas de alta freqüência contendo informações direcionais de detalhe na escala *j*, por isso

referenciadas como imagens de detalhe e $A_{2^j}^d f$ é a banda de baixa freqüência referenciada como uma imagem de baixa resolução na escala *j*.

Os filtros usados, H e G, para a decomposição *wavelet* no algoritmo de *Mallat* representam a função de base *Haar* para cor e textura, foram utilizados já no trabalho de Ramos et al. (2005), para a recuperação de imagens e agora neste trabalho é estendida aos vídeos.

Para ilustração, a Figura 4.6 apresenta os quatro canais de saída do algoritmo de decomposição *wavelet*, esses quatro conjuntos representam um conjunto para dados suavizados ou de baixa freqüência (aproximação), e mais três conjuntos direcionais de alta freqüência que são: horizontal, vertical e diagonal.

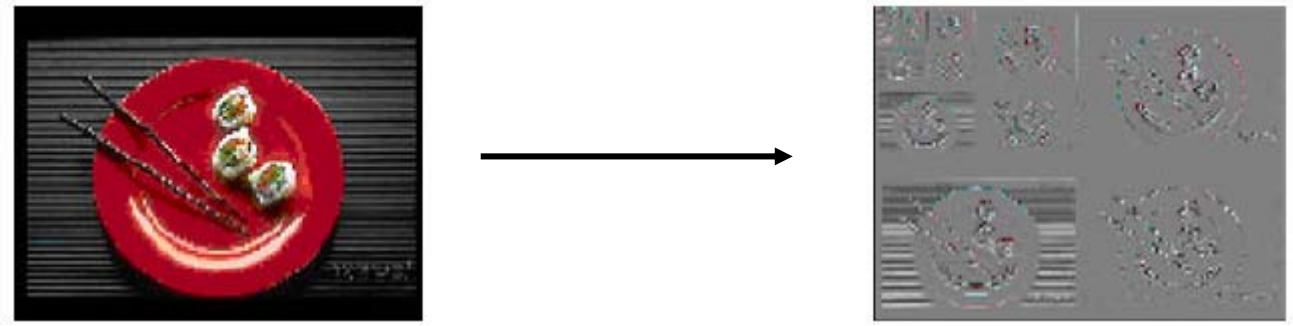


Figura 4.8 – Representação da decomposição wavelet em uma imagem

Na Figura 4.8, os conjuntos de saída da decomposição *wavelet* são para dados suavizados ou de baixa freqüência, $A_{2^j} f$ (A), os três conjuntos de detalhes, sendo de alta freqüência, vertical $D^1{}_{2^j} f$ (V), horizontal $D^2{}_{2^j} f$ (H) e diagonal $D^3{}_{2^j} f$ (D). Os conjuntos são subamostragens da imagem original, porém os coeficientes de detalhe contêm dados que somados a $A_{2^j} f$ reconstroem a imagem original. As imagens dos coeficientes vertical $D^1{}_{2^j} f$, horizontal $D^2{}_{2^j} f$ e diagonal $D^3{}_{2^j} f$, apresentadas na Figura 4.6 são representadas da seguinte forma: os coeficientes mais escuros representam as regiões de menor resposta aos filtros, enquanto que os coeficientes mais claros

representam as regiões de alta resposta do filtro.

No algoritmo *wavelet*, usando a base Haar, a dimensão dos canais de saída é a metade da dimensão da imagem original. Esta redução de dimensionalidade é realizada através da subamostragem da decomposição *wavelet.*

Cada um dos do quadro chave extraído é submetido a transformada wavelet conforme descrito no trabalho de Ramos et al. (2005).

4.6 FLUXO ÓTICO

O fluxo ótico é calculado de acordo com algoritmo descrito por Bouguet (2000), que consiste em primeiro lugar em representar a imagem na forma de uma pirâmide. Representemos a pirâmide de uma imagem I de tamanho $n_x \times n_y$. Seja $I^0 = I$ onde I^0 é a base da pirâmide. Essa imagem será a com maior resolução, a largura da imagem e sua altura neste nível são definidas por $n_x^0 = n_x$ e $n_y^0 = n_y$. A representação da pirâmide é construída computando-se I_1 a partir de I_0 e então computa-se I_2 a partir de I_1 e finalmente computa-se I_3 a partir de I_2;

Com L denotando o nível da pirâmide e I^L denotando a imagem correspondente ao nível L, temos então que $n_x^L = n_x$ e $n_y^L = n_y$ são respectivamente a largura e a altura de I^L. A imagem I^L pode ser definida na Equação 16 da seguinte forma:

$$
\begin{aligned}
I^L(x,y) = {} & \\
& \frac{1}{4} I^{L-1}(2x,2y) + \\
& \frac{1}{8}\left(I^{L-1}(2x,1,2y) + I^{L-1}(2x,1,2y) + I^{L-1}(2x,2y-1) + I^{L-1}(2x,2y+1)\right) + \\
& \frac{1}{16}\left(I^{L-1}(2x,1,2y-1) + I^{L-1}(2x,1,2y+1) + I^{L-1}(2x,1,2y+1) + I^{L-1}(2x,1,2y+1)\right)
\end{aligned}
\qquad (16)
$$

O segundo passo é encontrar características correspondentes, ou em outras palavras, um ponto u na imagem I e sua posição correspondente $v = u + d$ em uma imagem J tendo $d = \left[d_x, d_y \right]^T$ como o vetor de deslocamento deste ponto. Podemos assim descrever d como o vetor que minimiza a função definida a seguir:

$$\varepsilon(d) = \varepsilon(d_x, d_y) = \sum_{x=u_x-\omega_x}^{u_x+\omega_x} \sum_{y=u_y-\omega_y}^{u_y+\omega_y} \left(I(x,y) - J(x+d_x, y+d_y) \right)^2 \qquad (17)$$

Nesta pesquisa usou-se o algoritmo descrito por Bouguet que permite estimar a movimentação existente em uma seqüência de quadros. Neste artigo o fluxo ótico é obtido determinando-se pares correspondentes entre dois quadros em uma seqüência. O descolamento do ponto correspondente no par determina a intensidade do fluxo ótico e este valor é usado como referencia de movimentação para o quadro chave. (Bouguet, 2000)

4.7 RECUPERAÇÃO POR CONTEÚDO

Para conferir um significado a um conjunto de características, comparam-se as informações extraídas por uma função de similaridade. Nesse sentido, medidas de distância são usadas para computar a semelhança das características, onde o histograma pode ser usado como um conjunto ordenado de características (Smeulders et al., 2000).

Neste trabalho a medida de similaridade usada é a interseção de histogramas, mostrada na Equação 18 para as características estáticas e a distância euclidiana mostrada na Equação 19 para as características dinâmicas:

$$Dist\left(H, H^{'}\right) = \frac{\sum \min\left(h, h^{'}\right)}{\sum h'} \qquad (18)$$

h = histograma do quadro teste
h' = histograma do quadro testado

Um histograma é um vetor $[h_1, h_2, \ldots, h_n]$ em que cada entrada contém o número de *pixels* havendo a cor da imagem distribuída em proporções, e é considerada uma função de densidade de probabilidade das cores.

A distância euclidiana é utilizada como métrica para cálculo na medida de similaridade do movimento e é calculada sobre a média da movimentação do eixo X e do eixo Y na imagem através da análise da similaridade de distância euclidiana, que é dada por:

$$d_{Euclid} = \sqrt{\left(M\left(x, y\right) - M'\left(x, y\right)\right)^2} \qquad (19)$$

M(x,y) = valores do movimento X e Y no quadro teste
M'(x,y) = valores do movimento X e Y no quadro testado

Assim, quanto menor a distância euclidiana entre dois quadros, maior a semelhança de movimento entre elas.

A fim de obter-se a combinação dos resultados optou-se por ponderar os *scores* dados às imagens. Esta ponderação permite tornar uma determinada característica mais ou menos enfática na hora em que o sistema tenta recuperar um quadro chave. O score final é resultado da ponderação pela Equação 20, onde P_i^E é o fator de ponderação das características estáticas que são denotadas na equação por $Score_i^E$ e

P_i^D é o fator de ponderação das características dinâmicas representadas por $Score_i^D$, sendo $P_i^E + P_i^D = 1$.

$$Score_i = P_i^E \times Score_i^E + P_i^D \times Score_i^D \quad (20)$$

5 RESULTADOS EXPERIMENTAIS

Os filmes utilizados neste trabalho foram extraídos de DVDs de vários gêneros a fim de criar uma base de dados heterogênea, porém o áudio destes foi excluído deixando somente a parte visual presente. Como cada filme possui características bastante peculiares foi criada uma ficha que contém os detalhes relevantes sobre seu conteúdo estrutural.

5.1 DESCRIÇÃO DA BASE

Como não foi possível encontrar uma base publica de vídeos que fosse consenso na literatura, optou-se por definir uma base própria para este trabalho. A base principal é composta por cinco classes com um total de 208 quadros que representam 8 horas e meia de vídeo aproximadamente. Para cada quadro existe uma assinatura separada para as informações estáticas e dinâmicas devidamente extraídas conforme os métodos citados anteriormente nas seções 4.3 e 4.6.

A base é dividida em cinco classes, sendo elas: Águas, Brigas, Diálogos, Explosões e Perseguições, tais classes são mostradas a seguir mais detalhadamente. Esta base é gerada de forma a conter elementos heterogêneos e foi manualmente rotulada para que se pudesse comparar os resultados obtidos com o que realmente se espera da recuperação. Apesar disso a rotulação manual envolve a subjetividade humana que pode ser capaz de complicar a avaliação dos resultados.

A classe Águas representada na Figura 5.1 contém os quadros chaves que foram definidos como contendo elementos predominantemente ligados à água, como se vê temos cenas contemplando desde canais desaguando no mar (Figura 5.1a),

passando por córregos barrentos (Figura 5.1b) e em que ocorrem tomadas subaquáticas de mergulhos em mar aberto (Figura 5.1c) entre muitas outras.

(a) *(b)* *(c)*

Figura 5.1 - Exemplos da Classe Águas

Na classe Brigas exposta pela Figura 5.2, têm-se predominantemente tomadas onde há ataques corporais ou violência física principalmente entre pessoas. Como exemplares desta classe podemos observar lutas onde um homem e uma mulher, desferem golpes um contra o outro (Figura 5.2a), um adulto que espanca uma criança sem reação desta (Figura 5.2b) e um duelo de espadas travado por piratas dentro de um galpão (Figura 5.2c) dentre outras pertencentes a classe.

(a) *(b)* *(c)*

Figura 5.2 - Exemplos da Classe Brigas

No caso da classe Diálogos conforme a Figura 5.3, temos duas ou mais pessoas apenas conversando calmamente entre elas sem que haja muita dinâmica. Esta classe possui tomadas de pessoas conversando nas mais variadas ocasiões,

sejam em campo aberto (Figura 5.3a), ou dentro de um carro durante uma perseguição (Figura 5.3b) ou então na beira de uma piscina (Figura 5.3a).

(a) *(b)* *(c)*

Figura 5.3 - Exemplos da Classe Diálogos

A classe Explosões aqui expressada pela Figura 5.4, representa as tomadas em que veiculo, objetos ou construções explodem durante elas. Um fato bastante comum são as chamas que acontecem durante as explosões causadas por tiros (Figura 5.4a) ou colisões de veículos em alta velocidade (Figura 5.4c). Mas também há casos em que há explosões as chamas propriamente ditas, como no caso da Figura 5.4b onde há um curto circuito em um portão elétrico a explosão consiste em faíscas elétricas e fumaça somente.

(a) *(b)* *(c)*

Figura 5.4 - Exemplos da Classe Explosões

A última classe é a Perseguições vista na Figura 5.5 onde um elemento em fuga é seguido por outro, normalmente em alta velocidade. Neste caso usualmente

pessoas fogem a pé (Figura 5.5a), ou em veículos (Figura 5.5b), podendo ocorrer além de em diversos locais, como em corredores (Figura 5.5a), ou em locais abertos (Figura 5.5c) entre outros.

(a) (b) (c)

Figura 5.5 - Exemplos da Classe Perseguições

5.2 EXPERIMENTOS COM INFORMAÇÕES ESTÁTICAS

As informações obtidas pela decomposição *wavelet* do quadro é salva em um arquivo no formato *XML* para permitir sua análise tanto por parte humana quanto por parte computacional. As assinaturas são estruturadas por níveis na seguinte hierarquia:

- NIVEIS [0 – 2]
  - CANAIS [H, S,V]
    - PEDACO[(0,0) – (2,2)]
      - APROXIMACAO
      - HORIZONTAL
      - VERTICAL
      - DIAGONAL

Desta forma, abstraímos o quadro como sendo três imagens (devido aos três níveis de decomposição *wavelet*), cada nível tem seus três canais de cor separados e cada canal tem nove regiões. Por fim cada região possui então os valores obtidos dos coeficientes *wavelets*.

Os experimentos para as informações estáticas foram realizados conforme o trabalho de Ramos et al. (2005), mas utilizando a base de quadros obtida pela seleção de quadros chaves descrita na seção 4.2.

Nos parâmetros descritos neste trabalho a atual base foi testada e os resultados são mostrados na Figura 5.6.

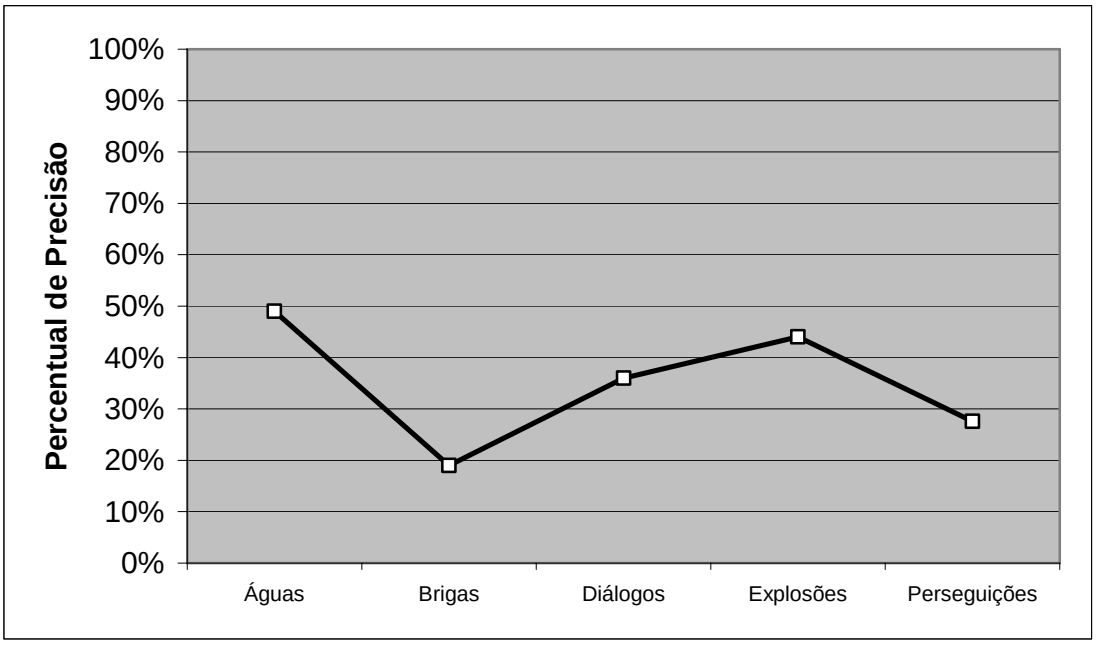


Figura 5.6 – Recuperação Utilizando Características Estáticas.

Neste experimento foram obtidos resultados cerca de 35% de acerto no caso da classe “Diálogos” e cerca de 10% para a classe “Perseguições”.

Além disso, podemos considerar que algumas classes como, por exemplo “Brigas” e “Diálogos” apenas visualmente, podem ser facilmente confundidas o que neste caso influenciou bastante os resultados. O mesmo também acontece entre as classes “Explosões” e “Perseguições” pois uma perseguição entre dois carros e um carro explodindo visualmente possuem informações idênticas até que as chamas predominem na cena.

5.3 EXPERIMENTOS COM INFORMAÇÕES DINÂMICAS

Para escolher qual a melhor forma de análise do movimento foram executados testes com quadros anteriores, posteriores ao quadro chave bem como a análise dos quadros anteriores e posteriores simultaneamente. Nesta análise a fim de garantir uma escolha ideal da variação de tempo a ser utilizada, cada um dos experimentos foi executado com 1,2,3,5,7 e 10 quadros em relação ao quadro chave.

As características de movimento foram extraídas de acordo como foi descrito no tópico 4.5 deste artigo e desta extração, foi gerada uma assinatura contendo as informações de movimento relativo ao quadro chave para cada uma das situações previstas anteriormente neste tópico.

Com base nestas assinaturas foram executados os testes de recuperação e o resultado destas recuperações foi analisado de forma a medir a qualidade das características testadas.

A Figura 5.7 demonstra o resultado de uma simulação de recuperação para três classes com movimentações distintas, feitas utilizando o somente o fluxo ótico referente aos quadros anteriores ao de referência nas variações de tempo citadas anteriormente.

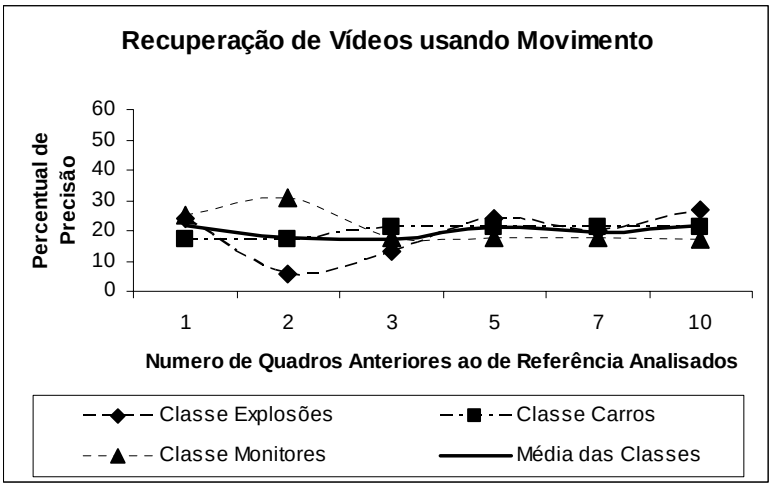


Figura 5.7 – Recuperação com 1,2,3,5,7,10 quadros

Nota-se nesse caso um desempenho bastante baixo, atingindo o máximo de 36% para a classe “Monitores” e menos de 6% para a classe “Explosões”. Tal resultado era previsível, pois conforme Rasheed e Shah um quadro chave é selecionado quando a diferença com seu anterior é muito grande. Sendo assim se o quadro chave não pertence à seqüência de quadros anterior a ele, a análise de movimento deste em relação aos quadros anteriores logicamente não retorna informações relevantes. (Rasheed et al., 2003)

Na Figura 5.8 se o resultado da mesma simulação, mas agora levando em conta a movimentação dos quadros anteriores e posteriores ao quadro chave na análise.

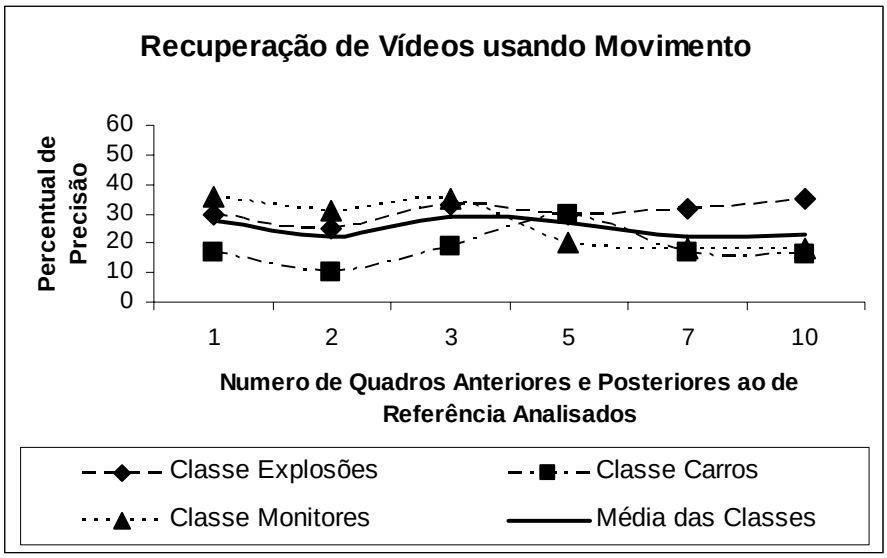


Figura 5.8 – Recuperação com 1,2,3,5,7,10 quadros

Usando essa configuração ainda sim, os resultados obtidos não foram os melhores conforme pode se ver na seção 5.3 onde os melhores resultados são obtidos, pois neste caso tal como o no caso anterior, a informação proveniente dos quadros anteriores ao quadro chave influem de forma negativa. Com isso os resultados apesar de baixos, foram mais significativos que os obtidos apenas o fluxo ótico referente aos

quadros anteriores ao de referência. Tais resultados se aproximam a 40% em vários momentos mas demonstra resultados de cerca de 11% em dados momentos.

Finalmente na Figura 5.9 temos o resultado desta análise utilizando apenas quadros posteriores ao de referência.

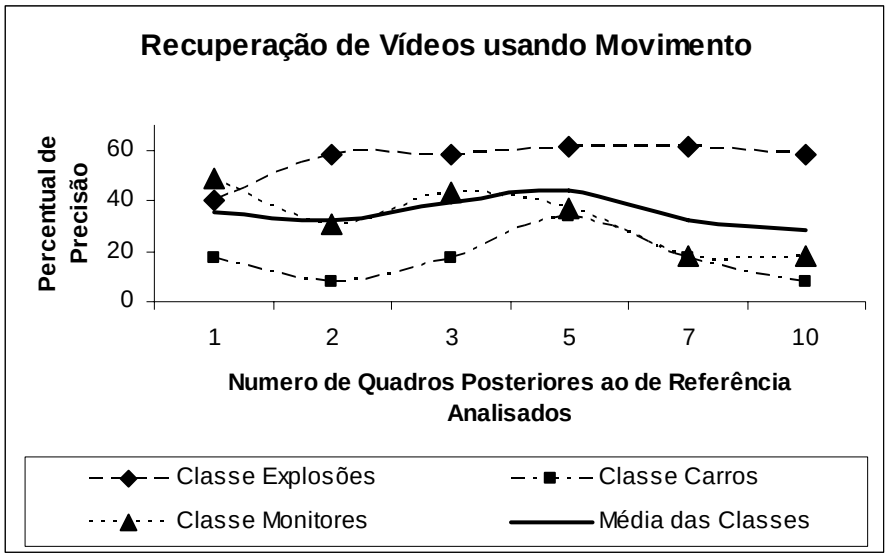


Figura 5.9 – Recuperação com 1,2,3,5,7,10 quadros

Com base nos resultados obtidos nessa etapa, as configurações dos experimentos de recuperação foram feitos pela análise do fluxo ótico referente a 5 quadros posteriores ao referenciado, pois foi neste ponto onde houve um melhor desempenho, demonstrando uma taxa de até 60% nos testes contra as taxas de cerca de 30% para os testes feitos com apenas quadros anteriores e de 40% para o uso dos quadros anteriores e posteriores.

Tendo apenas a referencia da direção em que o movimento resta ainda escolher a quantidade de quadros a serem analisados, desta forma com os resultados mostrados na Figura 5.4, a melhor alternativa é utilizar cinco quadros posteriores ao de referência. Cientes disto os testes passam a ser executados levando em consideração a informação do fluxo ótico extraído dos cinco quadros posteriores ao quadro chave.

Nestes parâmetros podemos analisar o impacto destas características escolhidas em toda a base de forma a comprovar a eficácia destas informações. O resultado do experimento realizado para analisar esta eficácia é mostrado na Figura 5.10.

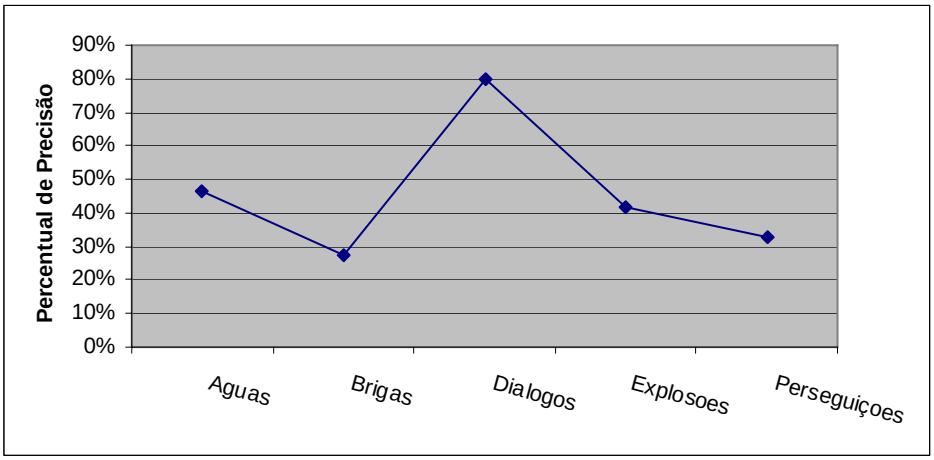


Figura 5.10 – Recuperação utilizando os parâmetros selecionados

Em posse destes dados notamos que foi possível atingir resultados de até 79%, mas mesmo assim algumas classes mantiveram índices de 12%.

Essas taxas são provenientes pela confusão gerada pela semelhança de movimentação, um exemplo disso é o caso de uma briga em que além da alta movimentação dos elementos da cena também há uma grande movimentação da câmera que filma a cena. O mesmo se dá em uma cena de perseguição, sendo assim em alguns casos esta semelhança prejudica os resultados.

Em contra partida a ambigüidade entre os diálogos e as brigas são solucionados em relação ao caso em que apenas se usa informações estáticas.

5.4 EXPERIMENTOS COM AMBAS AS INFORMAÇÕES

Utilizando os parâmetros retro-citados, foram realizados experimentos utilizando características estáticas e dinâmicas combinadas para descobrir a melhor ponderação entre as características. Para encontrar a melhor combinação foram testadas 11 configurações diferentes variando o valor de P_i^E entre 0 e 1 de 0,10 em 0,10 e tendo $P_i^D = 1 - P_i^E$. O resultado apresentado por este experimento e ilustrado pela Figura 12 leva a selecionar como melhor alternativa o uso de P_i^E = 0,3 e P_i^D =0,7, o que indica uma melhor ponderação com 70% do peso do *score* dado pelas características dinâmicas e 30% do peso do *score* sendo dado pelas características estáticas conforme se vê na Figura 5.11.

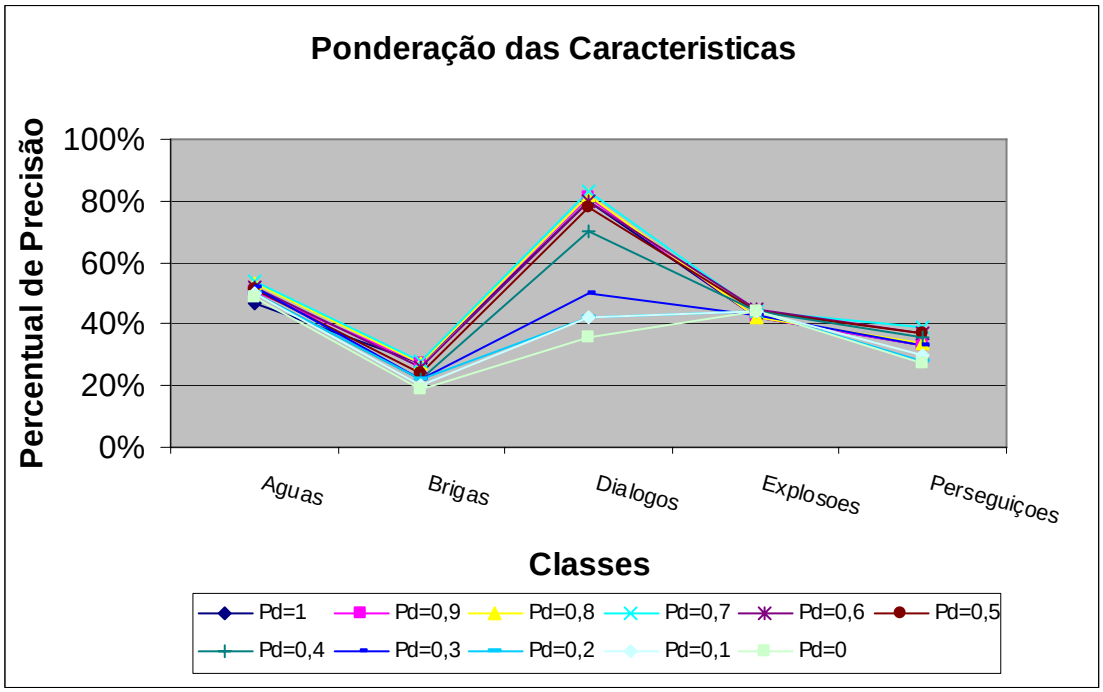


Figura 5.11 – Recuperação utilizando os parâmetros selecionados

Através do experimento anterior temos a melhor ponderação entre as características dinâmicas e estáticas, sendo assim agora podemos comprar os resultados obtidos na Figura 5.12.

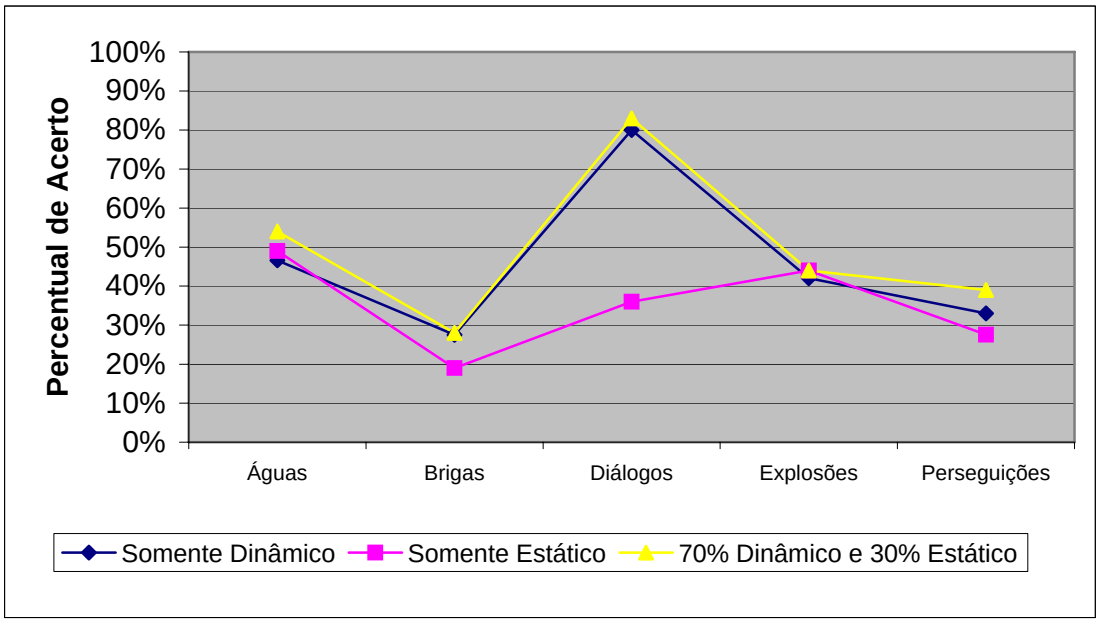


Figura 5.12 – Recuperação utilizando os parâmetros selecionados

Com essa ponderação é possível obter melhores resultados em todas as classes, embora em relação. Vale ainda ressaltar a grande melhora no caso da classe "Diálogos" em relação a apenas o uso das características estáticas.

Como forma de analisar o desempenho do método temos a matriz de confusão obtida na Tabela 3.

Tabela 3 – Matriz de Confusão

	Águas	*Brigas*	*Diálogos*	*Explosões*	*Perseguições*
Águas	***54%***	*14%*	*16%*	*0%*	*16%*
Brigas	*14%*	***28%***	*0%*	*35%*	*23%*
Diálogos	*16%*	*0%*	***83%***	*0%*	*1%*
Explosões	*0%*	*35%*	*0%*	***44%***	*21%*
Perseguições	*16%*	*23%*	*1%*	*21%*	***39%***

5.5 EXPERIMENTOS FINAIS E DISCUSÃO

Seguindo os passos descritos anteriormente foi executado um teste sobre a base a fim de se verificar visualmente a qualidade dos resultados obtidos pelo método.

Cada uma das figuras irá demonstrar os 12 primeiros resultados obtidos pelo método proposto para uma tomada de referência.

Foram executados cinco testes, um para cada classe, desta forma é possível visualizar os resultados em todas as classes da base de dados.

Figura 5.13 – Resultado da Recuperação na classe "Brigas" através da metodologia proposta.

Neste experimento há como referência uma tomada onde uma mulher e um homem lutam em um simulador, o método se mostrou capaz re retornar resultados pertinentes. Nota-se que as Figura 5.13(f), 13(g) e 13(i) apesar de não pertencerem à classe foram recuperadas, isto se dá à dinâmica muito grande e ordenada nos três casos. Enquanto a Figura 5.13(c) que possui chamas, na sua seqüência realmente faz parte de uma tomada de “Briga” bem como a 5.13(k) em que a briga acontece sobre uma espécie de jato, mas o visual que se tem em primeiro plano são as chamas das turbinas deste jato sobre o qual há uma briga. Outro caso interessante é o da Figura 5.13(h) onde a luta acontece durante um curto circuito que causa diversas alterações na iluminação, mas mesmo assim foi possível caracterizá-lo corretamente. Com isto podemos verificar que algumas classes visualmente distintas podem possuir dinâmicas muito parecidas.

Figura 5.14 – Resultado da Recuperação na classe “Diálogos” através da metodologia proposta.

Para a classe “Diálogos”, podemos notar uma menor confusão. Esta classe por possuir uma movimentação ínfima na grande maioria dos casos e pelo fato também de normalmente acontecer apenas entre pessoas torna-se bastante homogênea internamente. Nos casos em que a câmera se foca em apenas um dos personagens enquanto este fala, ainda sim se mantêm a baixa movimentação que na maioria das vezes é decorrente apenas das expressões faciais e alguns gestos.

Em casos de diálogos ocorridos em cenários abertos como campos ou ruas movimentadas, etc., o movimento do fundo é capaz de influir negativamente confundindo o sistema. Pelo fato de uma alta movimentação em um quadro chave ser muito incomum, este acaba então sendo confundido, muitas vezes, com a classe “Brigas” onde há também visualmente duas pessoas e uma alta movimentação.

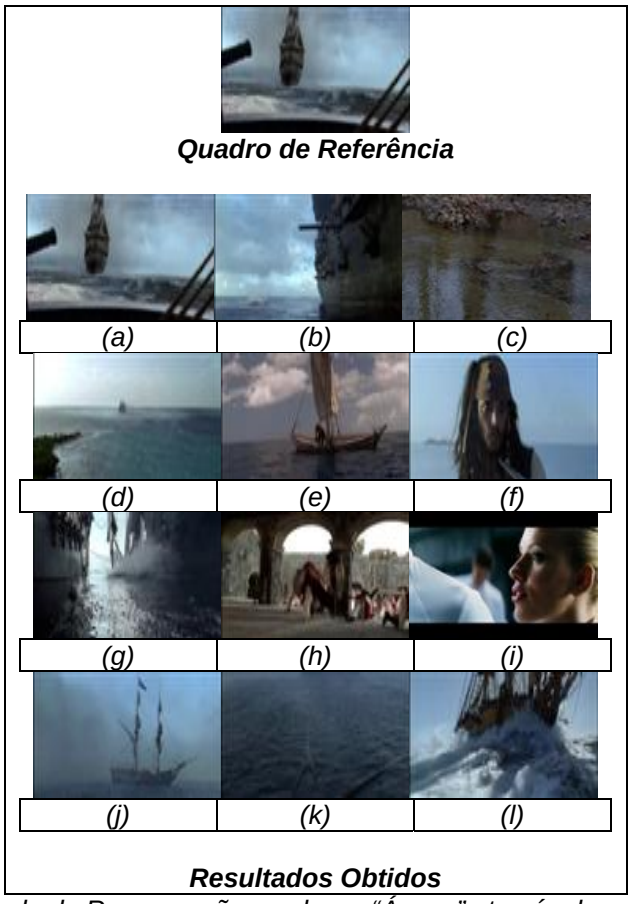


Figura 5.15 – Resultado da Recuperação na classe “Águas” através da metodologia proposta .

Na classe “Águas”, nós podemos notar duas classificações errôneas representadas nas Figuras 5.15(h) e 5.15(i), ambas foram classificadas desta maneira por possuírem uma movimentação baixa, predominante em regiões de fundo o que também ocorre no caso da água. Estas confusões também se dão em certos casos quando diálogos que também possuem baixa movimentação, se dão com o céu azul ao fundo, dando assim a “impressão” ao sistema que este céu na verdade é o mar, ou mesmo em casos como da Figura 5.15(i) onde a iluminação possui uma coloração azulada que visualmente leva o sistema a crer que se trata de mar.

Outro ponto a considerar é o caso da Figura 5.15(f) que apesar de em primeiro plano não parecer pertencer à classe, nota-se que ao fundo, em segundo plano há predominância do mar e no decorrer da tomada o pirata mergulha na água.

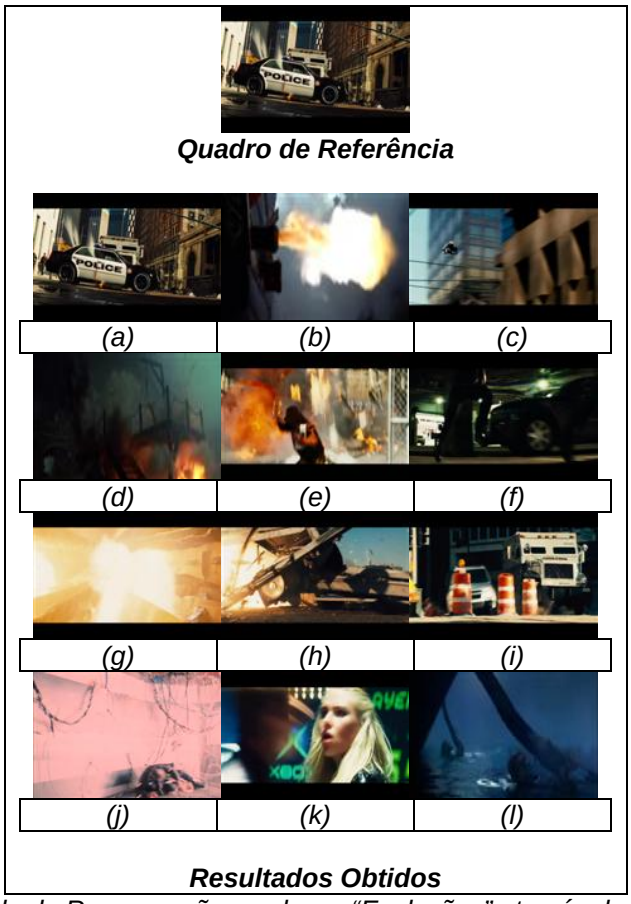


Figura 5.16 – Resultado da Recuperação na classe “Explosões” através da metodologia proposta.

Agora na classe “Explosões”, é interessante ressaltar o caso das tomadas representadas nas Figuras 5.16(a), 5.16(c), 5.16(i) onde apesar de não ocorrerem chamas no quadro chave, na seqüência da tomada ainda sim acontece uma explosão.

O fato de uma explosão se dar em vários quadros e não somente um, pode dificultar o sistema de através de características visuais identificar do que se trata a cena, mas devido ao movimento presente ocorrer em níveis elevados, as características dinâmicas possuem neste caso um papel muito importante, pois consegue suprir este papel e definir corretamente as tomadas.

Figura 5.17 – Resultado da Recuperação na classe “Perseguições” através da metodologia proposta.

Finalmente para a classe “Perseguições” acontecem algumas confusões com a classe explosões como no caso das Figuras 5.11(g) e 5.11(j). Nestes dois casos o que ocorre é um caso de perseguição em alta velocidade em que ao final tem-se uma explosão, desta forma ainda existem elementos dinâmicos da perseguição inseridos neste trecho e também em ambos os casos por ocorrerem em tomadas conseqüentes, há uma grande semelhança visual entre elas.

6 CONCLUSÃO

Para definir a similaridade em recuperação de informação, são desenvolvidos métodos que calculam a diferença entre quadros que estão sendo comparados. Como foi apresentada nesta dissertação, a tarefa de recuperar vídeos pode ser aplicada em diferentes áreas, tais como: engenharia e arquitetura; sistema de informação geográfica; em base de imagens médicas; dentre outras. Assim, muitos esforços foram direcionados em pesquisas e desenvolvimento dos CBVRS.

Este trabalho apresentou um método baseado em uma forma de indexação para recuperação de vídeos, levando-se em consideração a extração de características de cor e textura, combinados em um espaço *wavelet* juntamente com características de movimento representadas pelo fluxo ótico. Isso foi realizado com a inclusão de técnicas de recuperação de informação que permitiram tratar os vídeos de forma natural.

A escolha por essas características foi atribuída ao fato da cor representar a imagem como um todo, proporcionando informações relevantes para a tarefa de recuperação. Já a característica textura foi escolhida por estar intimamente relacionada com a característica cor, a qual foi tratada em diferentes níveis de multi-resolução. O movimento por sua vez, foi utilizado pelo fato do mesmo ser capaz de distinguir certas cenas que apesar de visualmente semelhantes possuem conteúdos muito distintos devido a sua dinâmica, como o caso de uma cena de um piquenique no campo e um jogo de futebol, por exemplo.

Levando em consideração os objetivos apresentados neste trabalho, concluiu-se que o modelo de cor HSV usado como forma de representar a cor das imagens tem a vantagem de ser um modelo que separa a intensidade da informação tonalidade e saturação, bem como, na relação que existe entre esses componentes que é muito próxima da forma na qual o homem percebe a cor.

A base Haar usada buscou realçar as melhores características das imagens trabalhadas, representando o sinal de forma redundante, pois não se sabia quais características eram mais significativas para se fazer a classificação, isso foi evidenciado através da extração das características, apresentada na Seção 4.3 e 4.6.

A associação da informação espacial ao atributo cor e textura foi realizada através da multi-resolução das imagens. Isso foi alcançado através da decomposição *wavelet* que permitiu a transformação das imagens em diferentes escalas.

A combinação do movimento com as características de cor e textura proporcionaram uma ponderação entre ambos os atributos de maneira a inserir a quantidade de informações necessária de cada uma das características para se obter os melhores resultados.

Os experimentos com a combinação, Seção 5.3 e 5.4, foram bem sucedidos com as métricas descritas na Seção 4.7, demonstrando taxas de até 83% e mantendo uma média geral de cerca de 50%.

A quantidade e a variedade de quadros que compõe a base de dados foi suficiente para fazer a classificação das imagens, bem como, validar as hipóteses levantadas durante os objetivos específicos.

6.1 TRABALHOS FUTUROS

Apesar do método proposto auxiliar na recuperação de informação baseada em conteúdo através da extração dos atributos cor e textura em espaço *wavelet* e do fluxo ótico, este trabalho não pretende esgotar as discussões sobre CBVRS. Isto porque o método desenvolvido neste trabalho considera um conjunto específico de características. Outros métodos podem ser propostos, quer incluindo novos atributos,

quer abordando as mesmas características, mas de maneira diferenciada, como por exemplo, a recuperação por similaridade através de pontos selecionados por regiões nas imagens.

Extensões possíveis da metodologia pode ser a expansão desta recuperação de informação para vídeos *on-line*, ou a avaliação de diferentes ponderações entre as características para as diferentes classes presentes na base.

REFERÊNCIAS BIBLIOGRÁFICAS

ADOBE, S. I. (2006). Premier Pro, Versão 2.0. Adobe Systems Incorporated.

ALSHUTH, P., HERMES, T., et al. (1998). "On video retrieval: Content analysis by ImageMiner." SPIE Storage and Retrieval for Image and Video Databases **3312**(236-247).

ARMAN, F., HSU, A., et al. (1994). "Image Processing on Encoded Video Sequences." Multimedia Systems **1**(5): 211-219.

BARRON, J. L., FLEET, D. J., et al. (1994). "Performance on optical flow techniques." nternational
Journal of Computer Vision **12**(1): 44-77.

BEAUCHENMIN, S. S. and BARRON, J. L. (1995). "The computation of optical flow." ACM Computing
Surveys **27**(3): 433-467.

BORECZKY, J. S. and ROWE, L. A. (1996). "Comparison of Video Shot Boundary Detection Techniques." Storage andRetrieval for Still Image and Video Databases IV(Proc. SPIE 2664): 170-179.

BOUGUET, J. Y. (2000). "Pyramidal Implementation of the Lucas Kanade Feature Tracker." Intel Corporation Microprocessor Research Labs.

BUENO, J. M., TRAINA, A. J. M., et al. (2002). "cbPACS: PACS com Suporte à Recuperação de Imagens Médicas Baseada em Conteúdo." VIII Congresso Brasileiro de Informática em Saúde **CBIS'2002**.

CANNY, J. (1986). "A Computational Approach to Edge Detection." IEEE Transactions on Pattern Analysis and Machine Intelligence **8**(6): 34-43.

CARDOSO, O. N. P. (2000). "INFOCOMP - Journal of Computer Science." Anais da III SECICOM **VOL.2,** (N.1): p. 33-38.

CASCIA, M. L. and ARDIZZONE, E. (1996). "Jacob: Just a content-based query system for video databases." IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP96), **'96**(May): 7-10.

CHUA, T. S. and RUAN, L. Q. (2001). "VRSS - A Video Retrieval and Sequencing System." ACM Transactions on Information Systems **13**(4): 373-407.

FACON, J. (1996). Morfologia Matemática: Teoria e Exemplos. Curitiba, Editora Universitária Champagnat da Pontifícia Universidade Católica do Paraná.

FERNEDA, E. (2003). **Recuperação de Informação: Análise sobre a comtribuição da Ciência da Computação para a Ciência da Informação**. Ciência da Informação e Documentação. São Paulo, USP Universidade de São Paulo. **Doutor:** 147.

GATTAS, M. (2005). "Introdução as Cores." TecGraf, Puc Rio: 14.

GONZALEZ, R. and WOODS, R. E. (2000). Processamento de Imagens Digitais.

HAFNER, J. L., SAWHNEY, H. S., et al. (1995). "Efficient Color Histogram Indexing for Quadratic Form Distance Functions." IEEE Transactions Pattern Analalisis **17**(7): 729-736.

HORN, B. K. P. (1986). "Robot Vision." Massachusetts Institute of Technology.

IBM. (2004). " IBM's Query by Image Content." from http://wwwqbic.almaden.ibm.com/.

KASTURI, R. and JAIN, R. (1991). "Dynamic Vision." Computer Vision: Principles, IEEE Computer Society Press.

LIENHART, R. (1999). "Comparison of automatic shot boundary detection algorithms." In SPIE Conf. on Storage and Retrieval for Image & Video Databases **3656**(VII): 290-301.

LIENHART, R., KUHMÜNCH, C., et al. (1997). "On the Detection and Recognition of Television Commercials." In Proceedings of the International Conference on Multimedia Computing and Systems: 509-516.

LITTLE, T. D. C., AHANGER, G., et al. (1993). "A Digital On- Demand Video Service Supporting Content-Based Queries." Proc. ACM Multimedia 93: 427-436.

LONG, H. and LEOW, W. (2000). "Perceptual texture space for content-based image retrieval." In Proc. Int. Conf. on Multimedia Modeling (MMM).

MAILLET, S. M. (2002). "Content-based Video Retrieval: An overview." from http://viper.unige.ch/~marchand/CBVR/.

MICROSOFT, C. (2000). Windows ME. M. Corporation.

MICROSOFT, C. (2001). Windows XP. M. Corporation.

MICROSOFT, C. (2004). Movie Maker. M. Corporation.

MILLS, T. J., PYE, D., et al. (2000). "Shoebox: A digital photo management system. ." Technical Report AT&T **2000**(10).

NAGASAKA, A. and TANAKA, Y. (1992). "Automatic Video Indexing and Full-Video Search for Object Appearances." Visual Database Systems II: 113-127.

NEBULASEARCH. (2005). "Content Based Image Retrieval." from http://www.nebulasearch.com/encyclopedia/article/Content-based_image_retrieval.html

OLIVEIRA, C. J. S., ARAUJO, A. A., et al. (2002). Proposta de um Protótipo de um Sistema de Recuperação de Imagens com Base na Cor. III Workshop em Tratamento de Imagens, Belo Horizonte, Minas Gerais.

OTTEWILL, M. and KALLENBACH, D. (1997). "Planet of Tunes." from http://www.planetoftunes.com/multi/index.html.

PASS, G., ZABIH, R., et al. (1996). "Comparing Images Using Color Coherence Vector." ACM Multimedia.

RAMOS, F., GOMES, H. M., et al. (2005). "Evaluating Content-Based Image Retrieval by Combining Color and Wavelet Features in a Region Based Scheme." CIARP: 679-690.

RASHEED, Z. and SHAH, M. (2003). "Scene Detection In Hollywood Movies and TV shows." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

SHAHRARAY, B. (1995). "Scene Change Detection and Content-Based Sampling of Video Sequences." Digital Video Compression: Algorithms and Technologies **Proc. SPIE 2419**: 2-13.

SHANON, X. J., BLACK, M. J., et al. (1998). "Summarization of Vídeo Taped Presentations:Automatic Analysis of Motion and Gesture." IEEE Transactions on Circuits And System for Vídeo Technology **xx**(y): 100-109.

SMEULDERS, A. W. M., WORRING, M., et al. (2000). "Content-Based Image Retrieval at the End of the Early Years." IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(12): 1349-1380.

SMITH, J. R. and CHANG, S. F. (1995). "Automated image retrieval using color and texture." Technical Report CU/CTR of Columbia University(July).

STRICKER, M. and DIMAI, A. (1996). "Color Indexing with Weak Spatial Constraints." SPIE Conference 2670.

SWANBERG, D., C.F.SHU, et al. (1993). "Knowledge Guided Parsing and Retrieval in Video Databases." Storage and Retrieval for Image and Video Databases, Proc. SPIE 1908: 173-187.

UEDA, H., T.MIYATAKE, et al. (1991). "IMPACT: An Interactive Natural-motion-picture Dedicated Multimedia Authoring System." proceedings of CHI, ACM New York **343-350**.

ULEAD, S. (2006). MediaStudio Pro, versão 8. Ulead Systems.

WIKIPEDIA. (2006). "Content Based Video Retrieval." from http://en.wikipedia.org/w/index.php?title=Content_Based_Video_Retrieval&oldid=56737580

ZABIH, R., MILLER, J., et al. (1995). "A Feature-Based Algorithm for Detecting and Classifying Scene Breaks." Proc. ACM Multimedia: 189-200.

ZHANG, H., KANKANHALLI, A., et al. (1993). "Automatic partitioning of full-motion video." Multimedia Systems **1**(1): 10-28.

7 APÊNDICE A

7.1 CÁLCULO DO FLUXO ÓTICO

```
#ifdef _CH_
#pragma package <opencv>
#endif

#ifndef _EiC
#include "cv.h"
#include "cxcore.h"
#include "highgui.h"
#include <stdio.h>
#include <math.h>
#endif

#define MAX_FEATURES 20

#define POS_X 192
#define POS_Y 264

CvSize Janela;
IplImage *src_image = 0, *dst_image = 0, *tmp_image=0;
IplImage *Frame1 = 0, *Frame2 = 0;
IplImage *x_image = 0, *y_image = 0;
IplImage *pyramid1 = 0, *pyramid2 = 0;

static const double pi = 3.14159265358979323846;

static double square(int a)
{
    return a * a;
}

int modulo(int a)
{
    return (int) sqrt(square(a));
}

static void opticflow(IplImage *frame1, IplImage *frame2)
{

    int number_of_features;
    CvPoint2D32f frame1_features[MAX_FEATURES];
    CvPoint2D32f frame2_features[MAX_FEATURES];
    char optical_flow_found_feature[MAX_FEATURES];
    float optical_flow_feature_error[MAX_FEATURES];
    CvSize optical_flow_window;
    CvTermCriteria optical_flow_termination_criteria;

    int i;
```

```
      int line_thickness;
      CvScalar line_color;
      CvPoint p,q;
      double angle;
      double hypotenuse;

      src_image = cvCloneImage( frame1 );
      dst_image = cvCloneImage( frame2 );
      tmp_image = cvCloneImage( frame1 );

      Frame1   = cvCreateImage( Janela, IPL_DEPTH_8U, 1);
      Frame2   = cvCreateImage( Janela, IPL_DEPTH_8U, 1);

      cvConvertImage(src_image, Frame1, 0);
      cvConvertImage(dst_image, Frame2, 0);

      x_image = cvCreateImage( Janela, IPL_DEPTH_32F, 1 );
      y_image = cvCreateImage( Janela, IPL_DEPTH_32F, 1 );


      number_of_features = MAX_FEATURES;
      cvGoodFeaturesToTrack(Frame1,    x_image,    y_image,    frame1_features,
&number_of_features, .01, .01, NULL);
      optical_flow_window = cvSize(3,3);
      optical_flow_termination_criteria  =  cvTermCriteria(  CV_TERMCRIT_ITER  |
CV_TERMCRIT_EPS, 20, .3 );

      pyramid1 = cvCreateImage( Janela,  IPL_DEPTH_8U, 1 );
      pyramid2 = cvCreateImage( Janela,  IPL_DEPTH_8U, 1 );


      cvCalcOpticalFlowPyrLK(Frame1,      Frame2,      pyramid1,      pyramid2,
frame1_features,  frame2_features,  number_of_features,  optical_flow_window,  5,
optical_flow_found_feature,                          optical_flow_feature_error,
optical_flow_termination_criteria, 0 );

      for(i = 0; i < number_of_features; i++)
      {

            if ( optical_flow_found_feature[i] == 0 )continue;

            line_thickness = 1;
            line_color = CV_RGB(255,0,0);

            p.x = (int) frame1_features[i].x;
            p.y = (int) frame1_features[i].y;
            q.x = (int) frame2_features[i].x;
            q.y = (int) frame2_features[i].y;


            angle = atan2( (double) p.y - q.y, (double) p.x - q.x );
            hypotenuse = sqrt( square(p.y - q.y) + square(p.x - q.x) );


            q.x = (int) (p.x - 3 * hypotenuse * cos(angle));
            q.y = (int) (p.y - 3 * hypotenuse * sin(angle));
```

```
            cvLine( tmp_image, p, q, line_color, line_thickness, CV_AA, 0 );

            p.x = (int) (q.x + 9 * cos(angle + pi / 4));
            p.y = (int) (q.y + 9 * sin(angle + pi / 4));
            cvLine( tmp_image, p, q, line_color, line_thickness, CV_AA, 0 );
            p.x = (int) (q.x + 9 * cos(angle - pi / 4));
            p.y = (int) (q.y + 9 * sin(angle - pi / 4));
            cvLine( tmp_image, p, q, line_color, line_thickness, CV_AA, 0 );

        }

        cvNamedWindow("Optical Flow", 1);
        cvShowImage("Optical Flow", tmp_image);

        cvWaitKey(1);
        cvReleaseImage(&src_image);
        cvReleaseImage(&dst_image);
        cvReleaseImage(&tmp_image);
        cvReleaseImage(&Frame1);
        cvReleaseImage(&Frame2);
        cvReleaseImage(&pyramid1);
        cvReleaseImage(&pyramid2);
        cvReleaseImage(&x_image);
        cvReleaseImage(&y_image);

    }
    int main (int argc, char **argv)
    {
        int inicio, fim;
        char nome[60];
        char ext[5];
        int contador;
        char im1[65], im2[65];

        CvCapture* capture = 0;
        capture = cvCaptureFromCAM( 0 );
        if( !capture )
        {
            fprintf(stderr,"Fonte Indisponivel...\n");
            return -1;
        }
        cvNamedWindow("Optical Flow", 1);

        IplImage *antigo = 0, *novo=0;
        IplImage *frame = 0;


        frame = cvQueryFrame ( capture );
        cvShowImage("Optical Flow", frame);
        Janela = cvGetSize( frame );
        cvMoveWindow("Optical Flow",0,0);
        int x;
        for (x=0;x<1000;x++)
        {
            IplImage *frame = 0;
              frame = cvQueryFrame ( capture );
              cvWaitKey(10);
```

```
            cvShowImage("Optical Flow", frame);
      }

      cvMoveWindow("Optical Flow",POS_X + 166,272);
      antigo = cvCloneImage( frame );
      cvShowImage("Optical Flow", frame);

      for (;;)
      {
            for (x=0;x<5;x++)
            {
                  IplImage *frame = 0;
                  frame = cvQueryFrame ( capture );
                  if( !frame )
                        break;

                  novo = cvCloneImage( frame );
                  opticflow( antigo, novo );
                  cvWaitKey(10);
                  cvReleaseImage(&novo);
            }
                  cvReleaseImage(&antigo);
                  antigo = cvCloneImage( frame );
      }

      cvDestroyAllWindows( );
      cvReleaseCapture ( &capture );

      return 0;
}

#ifdef _EiC
#endif
```

8 APÊNDICE B

8.1 SELEÇÃO DE QUADROS CHAVES

```
  #ifdef _CH_
  #pragma package <opencv>
  #endif

  #ifndef _EiC
  #include "cv.h"
  #include "cxcore.h"
  #include "highgui.h"
  #include <stdio.h>
  #include <math.h>
  #endif

  char nome[50];
  char *tmp;

  CvHistogram* Calcular_Histograma(IplImage *src)
  {

      if(src)
      {
          IplImage* h_plane = cvCreateImage( cvGetSize(src), 8, 1 );
          IplImage* s_plane = cvCreateImage( cvGetSize(src), 8, 1 );
          IplImage* v_plane = cvCreateImage( cvGetSize(src), 8, 1 );
          IplImage* planes[] = { h_plane, s_plane, v_plane };
          IplImage* hsv = cvCreateImage( cvGetSize(src), 8, 3 );
          int h_bins = 8, s_bins = 4, v_bins = 4;
          int hist_size[] = {h_bins, s_bins, v_bins};
          float h_ranges[] = { 0, 180 }; /* hue varies from 0 (~0°red) to 180
(~360°red again) */
          float s_ranges[] = { 0, 255 }; /* saturation varies from 0 (black-
gray-white) to 255 (pure spectrum color) */
          float v_ranges[] = { 0, 255 }; /* saturation varies from 0 (black-
gray-white) to 255 (pure spectrum color) */
          float* ranges[] = { h_ranges, s_ranges, v_ranges };
          int scale = 10;
          CvHistogram* hist=0;
          float max_value = 0;
          int h, s, v;

          cvCvtColor( src, hsv, CV_BGR2HSV );
          cvCvtPixToPlane( hsv, h_plane, s_plane, v_plane, 0 );
          hist = cvCreateHist( 3, hist_size, CV_HIST_ARRAY, ranges, 1 );

          cvCalcHist( planes, hist, 0, 0 );
          cvGetMinMaxHistValue( hist, 0, &max_value, 0, 0 );

          cvNamedWindow( "Source", 1 );
          cvShowImage( "Source", src );

           cvReleaseImage(&h_plane);
```

```
          cvReleaseImage(&s_plane);
          cvReleaseImage(&v_plane);
          cvReleaseImage(&hsv);
          return hist;

    }
}


float ComparaHistogramas(CvHistogram *h1, CvHistogram *h2, int Modo)
{

    float valor;

    valor = cvCompareHist(h1, h2, Modo);
    return valor;
}


float CompararImagens(IplImage *Imagem1, IplImage *Imagem2, int Modo)
{


    CvHistogram *hist1=0, *hist2=0;
    hist1 = Calcular_Histograma (Imagem1);
    hist2 = Calcular_Histograma (Imagem2);

    return ComparaHistogramas(hist1, hist2, Modo);
}

void SalvaImagens(IplImage *Imagem1, IplImage *Imagem2, int frame_n )
{
    char saidas[50];

    sprintf(saidas,"%s-%i.jpg", tmp, frame_n-1 );
    printf("-->%s\n",saidas);
    cvSaveImage(saidas, Imagem1);

    sprintf(saidas,"%s-%i.jpg", tmp, frame_n );
    printf("-->%s\n",saidas);
    cvSaveImage(saidas, Imagem2);

}
int main (int argc, char **argv)
{

    float x;
    int px,py,pxy;
    CvSize Tamanho_Imagem;
    IplImage *frame=0, *novo=0, *antigo=0, *shower=0;

    CvCapture* capture = 0;



    if( argc < 1 )
    {
```

```
            fprintf(stderr,"USE: Histogramas.exe %%1 %%2 %%3 %%4\n");

            fprintf(stderr,"     %%1 = Nome do Arquivo sem Extensao\n");
            fprintf(stderr,"     %%2 = Quadro Inicial do Video\n");
            fprintf(stderr,"     %%3 = Tipo de Comparacao\n");
            fprintf(stderr,"     %%4 = Limiar Maximo\n");
            return -1;
      }

      tmp = argv [1];
      sprintf(nome, "%s.avi", tmp);

      printf("%s\n", nome);
      printf("%s\n", tmp);
      capture = cvCaptureFromAVI( nome );

      if( !capture )
      {
            fprintf(stderr,"Fonte Indisponivel...\n");
            return -1;
      }


      cvSetCaptureProperty ( capture, CV_CAP_PROP_POS_FRAMES, atoi(argv[2]) );

      cvNamedWindow("Optical Flow", 1);

      frame = cvQueryFrame ( capture );

      cvShowImage("Optical Flow", frame);

      Tamanho_Imagem = cvGetSize(frame);

      py = Tamanho_Imagem.height;
      px = Tamanho_Imagem.width;
      pxy = px*py;

      printf("\nLargura = %i  -  Altura = %i    -   Pixels = %i\n\n", px, py,
px*py);

      antigo = cvCloneImage( frame );
      shower = cvCloneImage( frame );
      cvShowImage("Optical Flow", shower);

      int c;
      double p;

      for (;;)
      {
            IplImage *frame = 0;
            frame = cvQueryFrame ( capture );
            if( !frame )
                  break;

            p = cvGetCaptureProperty( capture, CV_CAP_PROP_POS_FRAMES );

            novo = cvCloneImage( frame );
```

```
            if(atoi(argv[3]) == 0)
            {
                  x = CompararImagens(antigo, novo, CV_COMP_CORREL);
                  if (x<atof(argv[4]))
                  {
                        printf("%i - Correlation  = %f\n", (int)p, x);
                        SalvaImagens( antigo, novo, (int)p);
                  }
            }
            if(atoi(argv[3]) == 1)
            {
                  x = CompararImagens(antigo, novo, CV_COMP_CHISQR);
                  if (x<atof(argv[4]))
                        printf("%f - Chi-Square   = %f\n", x/pxy, p);

            }
            if(atoi(argv[3]) == 2)
            {
                  x = CompararImagens(antigo, novo, CV_COMP_INTERSECT);
                  if (x<atof(argv[4]))
                        printf("%f - Intersection   = %f\n", x/pxy, p);
            }
            if(atoi(argv[3]) == 3)
            {
                  x = CompararImagens(antigo, novo, CV_COMP_CORREL);
                  if (x<atof(argv[4]))
                        printf("%i - Correlation  = %f\n", x, p);
                  x = CompararImagens(antigo, novo, CV_COMP_CHISQR);
                  if (x<atof(argv[4]))
                        printf("%i - Chi-Square   = %f\n", x/pxy, p);
                  x = CompararImagens(antigo, novo, CV_COMP_INTERSECT);
                  if (x<atof(argv[4]))
                        printf("%i - Intersection   = %f\n", x/pxy, p);
            }

            cvAbsDiff(novo,antigo,shower);
            cvShowImage("KeyFrames", shower);
                  cvReleaseImage(&novo);
                  cvReleaseImage(&antigo);
                  antigo = cvCloneImage( frame );

            c = cvWaitKey(10);
            if (c == 27)
                  break;
      }

      cvDestroyAllWindows( );
      cvReleaseCapture ( &capture );

      return 0;
}

#ifdef _EiC
#endif
```