Segmentação automática de tomadas em vídeo

Thiago Teixeira Santos

Dissertação apresentada

ao

Instituto de Matemática e Estatística

da

Universidade de São Paulo

para

obtenção do grau

de

Mestre em Ciência da Computação

Área de Concentração: **Ciência da Computação**

Orientador: **Prof^o Dr. Carlos Hitoshi Morimoto**

São Paulo – 2004

Segmentação automática de tomadas em vídeo

Este exemplar corresponde à redação final da
dissertação devidamente corrigida e defendida
por Thiago Teixeira Santos e aprovada pela
Comissão Julgadora.

São Paulo, 9 de agosto de 2004.

Banca Examinadora :

- Prof^o Dr. Carlos Hitoshi Morimoto (Orientador) - IME-USP
- Prof^o Dr. João Eduardo Kögler Jr. - POLI-USP
- Prof^o Dr. Roberto Marcondes Cesar Jr. - IME-USP

Agradecimentos

Inicialmente, agradeço a meu orientador, Prof. Dr. Carlos Hitoshi Morimoto, pelo esmero de sua orientação. É um professor ocupado mas que pacientemente recebia-me durante horas em sua sala, para conversas não só sobre o mestrado como também outros assuntos efêmeros diversos.

Gostaria de agradecer a todos os membros do antigo Laboratório de Imagens Visão e Estruturas Espaciais, em especial Fábio José Ayres, João Eduardo Kögler Jr. e Luciano Silva pelas muitas discussões produtivas, livros emprestados e muitos, muitos cafés.

É imprescindível agradecer a meu grande amigo Roberto Hirata Jr. pelas conversas, dicas, truques e inúmeras revisões das primeiras versões deste trabalho. Testei a paciência de Roberto até níveis incríveis. Felizmente, ela não se esgota.

Agradeço também aos eternos amigos da Ala Oeste do BCC IME-USP 1997, companheiros para todas as horas, e em especial a Rafael Caetano dos Santos, pela cuidadosa revisão deste texto.

Agradeço a meus pais, Duda e Bel, pelo eterno apoio logístico enquanto eu me dedicava a esta dissertação. E, sobretudo, ao enorme esforço de sustentar doze anos de ensino particular, cujos custos neste país desafiam mesmo a perseverança dos pais mais dedicados. Também agradeço a meu irmão, Breno, pela obtenção do famigerado artigo de Zhang *et al.*, referência obrigatória a qual, após três anos de busca, eu já havia desistido de obter.

E à Verônica que, como namorada, teve a paciência necessária com os caprichos deste mestrado.

A todos vocês, o meu muito obrigado.

Resumo

A área de recuperação de informação baseada em conteúdo visual vem ganhando importância graças ao volume de material visual existente (imagens e vídeo digitais), compartilhado e distribuído principalmente via Internet, e à capacidade de processamento alcançada pelos computadores pessoais na última década.

Novas formas de consumo, manipulação e exploração de vídeo digital podem ser criadas através da organização e indexação apropriada desse material. A *delimitação de tomadas* fornece uma base para a abstração e estruturação de vídeo, agregando quadros contíguos em seqüências de mesmo contexto, isto é, trechos com unidade em termos de tempo e espaço.

Nesta dissertação são apresentados os conceitos básicos de delimitação de tomadas e métodos tradicionais utilizados nesse tipo de segmentação, bem como vários resultados experimentais obtidos a partir de seqüências reais de TV. É analisada a distribuição das diferenças entre quadros sucessivos, calculada através de seus histogramas, na tentativa de caracterizar as transições entre tomadas e obter melhores parâmetros para a segmentação. Obtêm-se experimentalmente mais evidências que comprovam a superioridade da medida de intersecção de histogramas sobre outras medidas.

A principal contribuição do trabalho consiste no desenvolvimento de um algoritmo baseado no método *twin-comparison*, que apresenta melhor desempenho que o método original na detecção dos limites de tomadas por utilizar análise local da variação visual entre os quadros do vídeo.

Abstract

Visual content based information retrieval is an area of increasing importance due to the large volume of available material (digital images and videos), shared and distributed mainly by the Internet, and the processing power achieved by personal computer in the last ten years.

New ways to consume digital video and to manipulate and explore its visual information can be made by appropriately organizing and indexing this material. The *shot boundary detection* is a fundamental tool to video abstraction and structuring, combining near frames into sequences with similar context, segments with space and time unity.

This work presents the basic concepts about shot boundary detection, traditional methods used and several experimental results obtained from a real TV data set. The distribution of differences of neighboring frames, calculated from histogram comparison, is used to define the transitions between frames and to obtain better parameters for segmentation. Our experimental results show the superiority of the histogram intersection method over other measures.

Our main contribution is the development of a new algorithm based on the twin-comparison method, extended with local analysis of visual content variation between video frames. This algorithm was tested over hours of TV data, and performs better than the original method.

Sumário

1	Introdução	1
1.1	Estruturação e tomadas	2
1.2	Sumarização e indexação	4
1.3	Objetivos	6
1.4	Contribuições	8
1.5	Organização deste trabalho	8
2	Delimitação de tomadas	9
2.1	Delimitação de tomadas	9
2.2	Transições graduais cromáticas	10
2.2.1	<i>Fades</i>	11
2.2.2	Dissoluções	11
2.3	Transições graduais espaciais	13
2.4	Análise do problema	13
2.4.1	Fatores extremos	17
2.5	Avaliação de desempenho	18
2.5.1	Avaliação para cortes	18
2.5.2	Avaliação para transições graduais	19
3	Revisão bibliográfica	23
3.1	Características e métricas	23

3.1.1	Domínio espacial: comparação pixel-a-pixel	24
3.1.2	Domínio espacial: blocos	25
3.1.3	Domínio espacial: bordas	26
3.1.4	Domínio de cores: histogramas	27
3.2	Limiarização e detecção	29
3.2.1	<i>Twin-comparison</i>	30
3.2.2	Escala temporal e “platôs” de variação	31
3.2.3	Limiares adaptativos	33
3.2.4	<i>Step-variable</i>	34
3.3	Informação adicional sobre os quadros	36
3.4	Filtragem	41
3.5	Comparação entre métodos	42
4	Refinamentos	45
4.1	Distribuição das diferenças entre quadros adjacentes	45
4.2	Análise local	48
4.2.1	Reconhecimento: perda de cortes	51
4.3	Transições graduais	54
5	Resultados experimentais	59
5.1	Análise de parâmetros	59
5.1.1	Base para a distância relativa	60
5.1.2	Histogramas e número de <i>bins</i>	60
5.1.3	Separação de tomadas: o limiar k_{corte}^{ψ}	62
5.1.4	Transições graduais e o limiar k_{grad}	63
5.2	Testes	65
5.3	Aplicação: sumarização de vídeo digital	69
6	Conclusões	71

Lista de Figuras

1.1	Um sistema para recuperação de conteúdo em vídeo digital.	3
1.2	Estruturação de vídeo, como visto em Lienhart <i>et al.</i> [LPE97].	4
1.3	Uma transição gradual (dissolução) em <i>Airline Safety and Economy</i> (disponível via <i>The Open Video Project</i> [Pro]).	5
1.4	Tomada de <i>Airline Safety and Economy</i>	7
2.1	Montagem de um vídeo a partir de quatro tomadas.	10
2.2	Um corte em <i>Challenge at Glen Canyon</i> (disponível via <i>The Open Video Project</i> [Pro]).	10
2.3	Um <i>fade-out</i> seguido por um <i>fade-in</i> em <i>Airline Safety and Economy</i>	12
2.4	Função $\alpha(j)$ linear em um <i>fade</i>	13
2.5	<i>Wipe</i> no comercial de televisão <i>Cheerios/V-8 "Space Offer"</i> (disponível via <i>The Open Video Project</i> [Pro]).	14
2.6	Algoritmo 1.	16
2.7	Sinal da variação inter-quadros numa seqüência de tomadas S . Note como os intervalos de variação R e $\bar{R}$ afetam o desempenho da detecção [Han02]. Quanto maior a intersecção entre R e $\bar{R}$, maior o número de falsos-positivos e falsos-negativos.	16
2.8	Situação possível na avaliação do caso gradual. Uma transição proposta pode cobrir duas ou mais transições reais.	20

3.1	Composição de um código de cor para o cômputo eficiente de histograma de cor segundo Zhang <i>et al.</i> [ZKS93].	28
3.2	Gráfico $i \times z_h(q_i, q_{i+1})$ de um segmento de <i>Airline Safety and Economy</i>	29
3.3	Os cinco intervalos diferentes na Equação 3.19.	32
3.4	Gráfico $i \times z_h(q_i, q_{i+30})$ de um segmento de <i>Airline Safety and Economy</i>	33
3.5	Gráfico $i \times z_h(q_i, q_{i+1})$ de um segmento de <i>The Corvair in Action</i>	34
3.6	Os incrementos do algoritmo <i>Step-variable</i> de Xiong e Lee [XL98].	37
3.7	Gráfico $i \times \sigma^2(q_i)$ de um segmento de <i>Airline Safety and Economy</i> contendo três dissoluções.	39
3.8	Gráfico $i \times \mu(q_i)$ de um segmento de <i>Airline Safety and Economy</i> contendo <i>fades</i>	40
3.9	Gráfico $i \times \sigma^2(q_i)$ de um segmento de <i>Airline Safety and Economy</i> contendo <i>fades</i>	40
3.10	Gráfico $i \times \frac{d\sigma^2(q_i)}{di}$ de um segmento de <i>Airline Safety and Economy</i> contendo três dissoluções.	41
4.1	Distribuição de $z_{\cap}(q_i, q_{i+1})$ em CNN 03/02/1998.	46
4.2	Distribuição de $\psi_{\cap}^7(i)$ em CNN 03/02/1998.	49
4.3	<i>Boxplots</i> de $z_{\cap}^1(i)$ e suas várias filtragens, referente a CNN 03/02/1998. Os quartis permitem avaliar a grande interseção entre as distribuições inter e intra-tomadas.	50
4.4	Comparação entre os desempenhos na detecção de cortes (reconhecimento $\times$ precisão) obtidos pela aplicação de quatro filtros diferentes sobre $z_{\cap}^1(i)$	52
4.5	Uma seqüência de ação em uma chamada comercial da série de ficção científica <i>Babylon 5</i> em CNN 03/02/1998. Os cortes que delimitam as tomadas curtas anulam-se em $\psi_{\cap}^7(i)$	53

4.6	Falsos-negativos em CNN 03/02/1998, perdidos tomando-se $\psi_{\cap}^7(i) > 50\%$. Quadros escuros costumam levar a erros (a). Da mesma forma, distribuições de cores similares (b) ou com cenário comum (c) geram valores muito baixos em diferenças baseadas em histogramas.	54
4.7	Falso-positivo na detecção de transições graduais gerado por movimentação de um objeto de grandes dimensões em relação ao quadro.	57
5.1	Trecho do sumário construído automaticamente para ABC 19/06/1998. Referências (<i>web links</i>) ligam tomadas com conteúdo visual similar.	70

Lista de Tabelas

4.1	Resultados na detecção de cortes obtidos por limiares globais aplicados sobre $z_{\cap}^1(i)$ e calculados segundo a Equação 4.1 em CNN 03/02/1998.	47
4.2	Resultados na detecção de cortes obtidos por limiares globais sobre $\psi_{\cap}^7(i)$ em CNN 03/02/1998.	48
4.3	Resultados na detecção de cortes obtidos por limiares globais sobre $\psi_{\cap}^7(i)$ em CNN 03/02/1998 após a remoção manual de tomadas com comprimento menor do que 8 quadros.	52
4.4	Resultados obtidos pelo Algoritmo 4 para $m = 7$	56
4.5	Resultados obtidos pelo Algoritmo 4 para $m = 3$	57
5.1	Conjunto de testes, fornecido pelo LDC/NIST para o TRECVID'04, utilizado neste trabalho. A coluna <i>Fades</i> se refere a <i>fades-out</i> seguidos imediatamente por <i>fades-in</i>	60
5.2	Resultados obtidos pelo Algoritmo 4 com $k_{\text{corte}} = 50\%$ para vários valores de m	61
5.3	Resultados obtidos pelo Algoritmo 4 com $k_{\text{corte}} = 50\%$ para vários valores de m	61
5.4	Resultados obtidos pelo Algoritmo 4 sobre $z_{\cap}^1(i)$ com $k_{\text{corte}}^{\psi} = 50\%$, $k_{\text{grad}} = 0.04$ e $m = 11$. Foram utilizados histogramas com 64 e 256 <i>bins</i>	62
5.5	Resultados obtidos pelo Algoritmo 4 sobre $z_{\circ}^1(i)$ com $k_{\text{corte}}^{\psi} = 50\%$, $k_{\text{grad}} = 0.04$ e $m = 11$. Foram utilizados histogramas com 64 e 256 <i>bins</i>	62
5.6	Resultados obtidos pelo Algoritmo 4 sobre $z_{\cap}^1(i)$ com $k_{\text{grad}} = 0.04$, $m = 11$ e 64 <i>bins</i> para vários valores de k_{corte}^{ψ}	63

5.7	Resultados obtidos pelo Algoritmo 4 sobre $z_o^1(i)$ com $k_{\text{grad}} = 0.04$, $m = 11$ e 64 <i>bins</i> para vários valores de k_{corte}^ψ	64
5.8	Resultados obtidos pelo Algoritmo 4 sobre $z_{\cap}^1(i)$ com $k_{\text{corte}}^\psi = 50\%$, $m = 11$ e 64 <i>bins</i> para vários valores de k_{grad}	65
5.9	Resultados obtidos pelo Algoritmo 4 sobre $z_o^1(i)$ com $k_{\text{corte}}^\psi = 50\%$, $m = 11$ e 64 <i>bins</i> para vários valores de k_{grad}	66
5.10	Resultados obtidos pelo Algoritmo 4 sobre $z_{\cap}^1(i)$ com $m = 11$ no conjunto de testes oriundo do canal de televisão CNN.	67
5.11	Resultados obtidos pelo Algoritmo 4 sobre $z_{\cap}^1(i)$ com $m = 11$ no conjunto de testes oriundo do canal de televisão ABC.	68

Capítulo 1

Introdução

Os constantes avanços em poder de processamento, capacidade de armazenamento, memória e largura de banda tornaram possível ao usuário de computadores o acesso, a manipulação e a edição de áudio, imagem e vídeo digitais. Este número elevado de usuários aptos ao consumo de tais mídias vem despertando o interesse da academia e da indústria.

Em 2001, o *National Institute of Standards and Technology* (NIST), patrocinador das *Text Retrieval Conferences* (TRECs), iniciou uma nova modalidade em sua conferência: a *TREC Video Retrieval Evaluation* (TRECVID). O novo ramo se destina à avaliação de técnicas de recuperação de informação contida em vídeo digital como, por exemplo, encontrar segmentos que contenham pessoas, uma determinada face conhecida, paisagens naturais, paisagens urbanas ou música instrumental em sua seqüência de áudio.

O TRECVID conta com o apoio da rede de TV BBC. As empresas de televisão têm claro interesse em tecnologias emergentes como TV interativa e vídeo por demanda, além do reaproveitamento de suas bases de material audiovisual acumuladas por anos. São exemplos os sistemas para navegação em noticiários apresentados por Zhang *et al.* [ZSW95], Bertini *et al.* [BDBP02] e pela AT&T [CHH⁺00].

Novas aplicações em vídeo digital, como busca e navegação, demandam uma nova estrutura para esta mídia. Antigamente apenas disponível em fitas, o vídeo só possibilitava acesso seqüencial. Um aparelho de vídeo-cassete só é capaz de “navegar” pelo material avançando

ou retrocedendo a fita. Com o armazenamento do vídeo em computadores, formas mais sofisticadas de acesso se tornaram possíveis. O *Moving Picture Experts Group* (MPEG), ao criar o padrão MPEG-4 [MPEG02], em 1998, apresentou uma decomposição do conteúdo audiovisual em *objetos de mídia* que podem ser combinados e reutilizados de formas diferentes em aplicações distintas.

Posteriormente, em 2001, o grupo apresentou seu novo padrão MPEG-7 [MPEG03], formalmente chamado *Multimedia Content Description Interface*. O padrão acrescenta mecanismos de anotação para descrever formas, cores, texturas e movimento presentes no vídeo. Mais do que isso, o MPEG-7 permite a estruturação de diversos elementos espaço-temporais em relações de antes/depois, frente/atrás, acima/abaixo, parte/todo.

Para possibilitar novas formas em seu envio, consulta e consumo, o vídeo digital deve estar *estruturado*. Considere um sistema de busca em acervos de vídeo digital, como mostrado na Figura 1.1. A estruturação incrementa a eficiência dos extratores de características pois estes não necessitam mais efetuar processamento quadro-a-quadro. A própria estrutura em si pode conter também características de interesse, para responder questões como:

- *Quais os segmentos que compõem a primeira cena do filme?*
- *Onde mais se encontra este personagem?*
- *Quais os segmentos filmados neste cenário?*

Assim, torna-se necessário definir uma unidade para a decomposição do vídeo que leve a uma estrutura útil às várias possíveis aplicações.

1.1 Estruturação e tomadas

Uma *tomada*¹ consiste de um ou mais quadros, gerados e gravados de forma contígua, representando uma ação contínua em relação a tempo e espaço [DSP91]. Cenas são obtidas

¹Utilizaremos “tomada” neste trabalho como tradução do termo “*shot*”.

Figura 1.1: Um sistema para recuperação de conteúdo em vídeo digital.

pela combinação de várias tomadas e delas obtem-se o vídeo completo, como exibido na Figura 1.2. A *transição* de uma tomada para a seguinte pode ser feita através da simples concatenação de quadros ou da aplicação de efeitos mais complexos de edição, como dissoluções e *fades*, que sobrepõem segmentos das duas tomadas na forma de uma passagem gradual de uma para a outra, como visto na Figura 1.3.

As tomadas são as estruturas com a mais fina granularidade em relação à unidade de ação (dinâmica de objetos em determinado espaço durante um intervalo de tempo), sendo a base para a maioria dos sistemas de segmentação e sumarização de vídeo [LPE97, Han02]. De modo similar a decomposição de texto em seções, parágrafos e frases, a divisão de vídeo em tomadas pode ser utilizada para a indexação de vídeo e construção de sumários, promovendo formas eficientes de busca e navegação [DSP91, SLS03].

Além de navegação e busca, várias outras aplicações podem se beneficiar da estruturação por tomadas. Ferramentas automáticas e semi-automáticas para colorização de vídeos em preto e branco podem associar mapas de cores distintos a tomadas diferentes, agilizando o trabalho dos editores. A restauração de vídeos antigos pode ser auxiliada de forma semelhante, reaproveitando conteúdo e configurações entre tomadas similares [Han02].

Figura 1.2: Estruturação de vídeo, como visto em Lienhart *et al.* [LPE97].

1.2 Sumarização e indexação

Grandes quantidades de informação, como as disponíveis em vídeo (e em acervos de vídeo) necessitam de técnicas eficientes de navegação e busca. A navegação exige uma exibição sucinta e representativa do conteúdo do material, como os sumários utilizados em livros e revistas. Já a busca se beneficia da construção de índices que referenciem de modo apropriado os elementos desejados.

Uma forma de representar segmentos do vídeo é a utilização de um ou mais *quadros-chaves*. Considere como exemplo a Figura 1.4. Se desejamos obter um sumário visual conciso do vídeo em questão, podemos admitir que há grande redundância entre os quadros da seqüência, de modo que um único deles poderia representar de modo satisfatório todo o conteúdo do segmento para fins de navegação.

Devido a sua unidade em termos de tempo e espaço, uma tomada é adequada não só à procura de quadros chaves como também para a construção de *mosaicos*. Mosaicos gerados por técnicas como *strip pasting*, a partir dos quadros de uma tomada, podem representar, em uma única imagem, toda a ação contida em um segmento do vídeo [BDF⁺99].

Quadros-chaves e mosaicos podem ser utilizados na criação de índices, através da extração

Figura 1.3: Uma transição gradual (dissolução) em *Airline Safety and Economy* (disponível via *The Open Video Project* [Pro]).

de características de imagem, como cor, textura e forma. Por sua vez, as tomadas são adequadas para a extração de características envolvendo movimento.

Sistemas complexos para problemas de detecção de padrões, como reconhecimento de faces [ZCPR03], podem se beneficiar dessas representações sucintas do conteúdo e ter seus desempenhos incrementados, pois podem operar em uma massa menor de dados, como visto anteriormente na Figura 1.1.

1.3 Objetivos

O principal objetivo do presente trabalho é o estudo e desenvolvimento de algoritmos para segmentação e estruturação de vídeo digital. Em particular, dado seu papel central na abstração de vídeo, será tratado o problema da *delimitação de tomadas*².

Segundo Hanjalic [Han02], dois pontos são essenciais quanto à robustez de um delimitador de tomadas:

1. bom desempenho nas principais variedades de transição entre tomadas e
2. desempenho similar ao operar em vídeos diferentes, com um mínimo de ajuste para cada vídeo em questão.

Considerando os vários efeitos de edição utilizados na concatenação de tomadas, os movimentos de objetos em cena e das próprias câmeras, além das mudanças de iluminação, tem-se uma idéia da dificuldade em atingir resultados satisfatórios de acordo com os critérios acima.

Trabalhamos na busca de *algoritmos adaptativos* que permitam a delimitação das tomadas e que obtenham bom desempenho independente da natureza do vídeo ou do trecho de vídeo em questão. Tal idéia está presente nos trabalhos de Zhang *et al.* [ZKS93], Ferman e Tekalp [FT98] e Hanjalic [Han02], entre outros.

²Conhecido na literatura como *shot boundary detection* (SBD).

Figura 1.4: Tomada de *Airline Safety and Economy*.

1.4 Contribuições

As principais contribuições deste trabalho são:

- um amplo levantamento bibliográfico, com o estudo dos principais métodos que compõem o estado da arte no que se refere à detecção de transições entre tomadas de vídeo;
- realização de vários experimentos comparativos;
- o desenvolvimento de um algoritmo adaptativo de delimitação de tomadas, que visa atingir os critérios de robustez apresentados na Seção 1.3, e
- de um método para a composição automática de sumários de vídeo para exibição na *Web*.

1.5 Organização deste trabalho

O presente texto está organizado da seguinte forma: o Capítulo 2 descreve os tipos mais comuns de transições em edição de vídeo e o problema de delimitação de tomadas, seguido pela apresentação de algumas das principais soluções da literatura no Capítulo 3. O Capítulo 4 discorre sobre melhorias possíveis dessas soluções e descreve nosso algoritmo adaptativo para delimitação de tomadas. O conjunto de testes e os resultados obtidos nos experimentos são descritos no Capítulo 5. O Capítulo 6 encerra o trabalho com algumas conclusões.

Capítulo 2

Delimitação de tomadas

Delimitar as tomadas em uma seqüência de vídeo é um passo importante para sua estruturação e para a análise de eventos. A passagem de uma tomada para outra representa uma troca de contexto essencial se estamos interessados, por exemplo, na construção de uma imagem panorâmica, no estudo da trajetória de um objeto ou qualquer outra análise dependente de unidade temporal e/ou espacial.

Este capítulo apresenta algumas das formas mais comuns de *transição* entre duas tomadas consecutivas e uma formalização do problema de delimitação de tomadas. Prossegue com uma análise do problema e métricas para avaliação da qualidade de soluções propostas por um delimitador automático.

2.1 Delimitação de tomadas

Um vídeo digital é uma seqüência $V = \langle q_1, q_2, \dots, q_n \rangle$ composta por quadros q_i , imagens de resolução $M \times N$. É obtido pela montagem de várias tomadas em um processo de edição, onde cada par consecutivo de tomadas é separado por uma *transição*. Pode-se representar uma transição T_i entre duas tomadas consecutivas S_i e S_{i+1} por um par de índices (s, t) , $s < t$, tais que $S_i = \langle \dots, q_{s-1}, q_s \rangle$ e $S_{i+1} = \langle q_t, q_{t+1}, \dots \rangle$, como exibido na Figura 2.1.

A forma mais simples de transição entre duas tomadas consecutivas é o *corte*, onde

$t = s + 1$, ilustrado na Figura 2.2. Já as *transições graduais* ocorrem quando $t > s + 1$, onde os quadros no intervalo entre s e t são *editados*, formados por uma composição dos quadros originais das duas tomadas em questão. É o que ocorre entre as tomadas S_2 e S_3 na Figura 2.1.

Figura 2.1: Montagem de um vídeo a partir de quatro tomadas.

Figura 2.2: Um corte em *Challenge at Glen Canyon* (disponível via *The Open Video Project* [Pro]).

2.2 Transições graduais cromáticas

Transições graduais obtidas pela manipulação das cores ou da intensidade luminosa dos quadros de duas tomadas sucessivas são chamadas *transições cromáticas* [HWJ94]. Nesta

categoria, os efeitos de edição mais largamente utilizados são os *fades* e as *dissoluções*.

2.2.1 *Fades*

Um *fade-out* consiste na diminuição progressiva da luminosidade nos quadros de uma tomada S_i até a obtenção de quadros completamente negros¹. Inversamente, um *fade-in* é o aumento progressivo da luminosidade, a partir de quadros negros, até a visualização da tomada, com sua luminosidade natural. *Fades-out* seguidos por *fades-in* são muito comuns em edição de vídeo, como visto na Figura 2.3.

Considere uma tomada $S_i = \langle a_1, a_2, \dots, a_m \rangle$. Um *fade-out* insere no vídeo V quadros modificados de S_i segundo a forma

$$q_j = \alpha(j) \cdot a_j. \quad (2.1)$$

A função $\alpha(j)$ controla o quanto da luminosidade do quadro original será utilizada no quadro editado correspondente², sendo monotonicamente decrescente no intervalo entre s e t (Figura 2.4).

Já um *fade-in*, de modo inverso, gera quadros modificados da forma

$$q_j = (1 - \alpha(j)) \cdot a_j. \quad (2.2)$$

2.2.2 Dissoluções

As dissoluções são generalizações dos *fades*. Considere duas tomadas $S_i = \langle a_1, a_2, \dots, a_m \rangle$ e $S_{i+1} = \langle b_1, b_2, \dots, b_n \rangle$. Uma dissolução entre S_i e S_{i+1} gera uma seqüência de quadros editados entre s e t definidos por

$$q_j = \alpha(j) \cdot a_j + (1 - \alpha(j)) \cdot b_j. \quad (2.3)$$

¹Ou, como caso geral, quadros monocromáticos.

²A Equação 2.1 é uma simples multiplicação de escalar por matriz, esta última relativa à componente do quadro associada à luminosidade (V em HSV ou L em La*b*, por exemplo).

Figura 2.3: Um *fade-out* seguido por um *fade-in* em *Airline Safety and Economy*.

Figura 2.4: Função $\alpha(j)$ linear em um *fade*.

Em outras palavras, na dissolução há um *fade-out* em S_i ocorrendo simultaneamente a um *fade-in* em S_{i+1} . Dissoluções têm o aspecto ilustrado anteriormente na Figura 1.3.

2.3 Transições graduais espaciais

Ao invés de utilizar alterações de luminosidade ou cor, algumas transições atuam no domínio espacial. Seu representante mais comum é o *wipe*, onde os quadros da tomada anterior podem “deslizar” sobre os quadros da tomada posterior, ou ainda gerar um efeito de “paginação”. Em outras palavras, regiões dos quadros de uma tomada são substituídos pelas regiões equivalentes da tomada seguinte, como ilustrado na Figura 2.5.

O presente trabalho terá como foco a detecção de cortes, dissoluções e *fades*, uma vez que esses são os efeitos mais largamente utilizados em edição de vídeo.

2.4 Análise do problema

Hanjalic [Han02] apresenta uma análise do problema de delimitação de tomadas. A base para a resolução do problema reside no fato que quadros em torno de uma transição geralmente apresentam uma mudança significativa em seu conteúdo visual. O processo de detecção consiste no reconhecimento de *descontinuidades* no fluxo do conteúdo visual da

Figura 2.5: *Wipe* no comercial de televisão *Cheerios/V-8 “Space Offer”* (disponível via *The Open Video Project* [Pro]).

seqüência de vídeo.

Toma-se uma *característica* do conteúdo visual dos quadros que será utilizada como base para o método, junto a uma *métrica* para mensurar a variação $z^l(i) = z(q_i, q_{i+l})$ dessa característica entre dois quadros q_i e q_{i+l} , onde $l \geq 1$ é uma distância inter-quadros. Cada $z^l(i)$, $1 \leq i \leq n$, é comparado por um *detector* a um *limiar* $K(i)$, decidindo-se assim se q_i consiste ou não em um *limite* de uma tomada. O algoritmo a seguir sumariza o processo:

Algoritmo 1 (Determinação de descontinuidades) *Dada uma seqüência de vídeo $V = \langle q_1, q_2, \dots, q_n \rangle$, encontra uma seqüência de transições $\mathcal{T} = \langle T_1, T_2, \dots, T_v \rangle$ em V .*

D1. [Inicialize.] Atribua $i \leftarrow 1$, $\mathcal{T} \leftarrow \emptyset$.

D2. [Calcule a variação.] Faça o processamento adequado dos quadros q_i e q_{i+l} e calcule $z^l(i)$.

D3. [Obtenha o limiar.] Calcule $K(i)$.

D4. [Excedeu?] Se $z^l(i) > K(i)$, adicione a nova transição $(i, i+l)$ a $\mathcal{T}$ e atribua $i \leftarrow i+l+1$.
Vá para D6.

D5. [Incremente i .] Atribua $i \leftarrow i + 1$.

D6. [Fim do vídeo?] Se $i \geq n$, o algoritmo termina. Caso contrário, volte a D2. ■

Veremos no Capítulo 3 como a maioria dos algoritmos apresentados pela literatura são variações do Algoritmo 1. Cada algoritmo assume certos valores para parâmetro l e define as funções $z(q_i, q_j)$ e $K(i)$.

Para o sucesso do Algoritmo 1, o par característica/métrica utilizados em $z^l(i)$ deve ser tão discriminante quanto possível. Seja $\bar{R}$ o intervalo mínimo de valores de variação tal que $z_l(i) \in \bar{R}$ para todo i onde q_i e q_{i+l} estão dentro de uma mesma tomada. Analogamente, seja R um intervalo mínimo de valores tal que $z_l(i) \in R$ para todo i onde q_i e q_{i+l} estão em tomadas diferentes. A função $z_l(i)$ é tão discriminante quanto menor for a interseção entre R e $\bar{R}$, como indicado na Figura 2.7. Uma interseção torna-se fonte de *falsos positivos* e *falsos negativos* durante a fase de detecção, degenerando a qualidade da solução.

Figura 2.6: Algoritmo 1.

Figura 2.7: Sinal da variação inter-quadros numa seqüência de tomadas S . Note como os intervalos de variação R e $\bar{R}$ afetam o desempenho da detecção [Han02]. Quanto maior a intersecção entre R e $\bar{R}$, maior o número de falsos-positivos e falsos-negativos.

2.4.1 Fatores extremos

As maiores diferenças no conteúdo visual de quadros em uma mesma tomada são causadas principalmente por movimentações de objetos, movimentação da câmera e mudanças de iluminação. Características e métricas muito sensíveis a esses eventos terão uma grande sobreposição dos intervalos R e $\bar{R}$ ($\bar{R}$ se expande). Se $z^l(i)$ pudesse ser construída sobre características e métricas robustas em relação a esses fatores, não só a interseção $R \cap \bar{R}$ seria pequena, mas também haveria intervalos *consistentes* mesmo em vídeos de naturezas diferentes. Um filme de ação e um vídeo *clip* têm níveis maiores de movimentação e alteração de luminosidade do que um drama ou uma entrevista. Dizer que os intervalos são consistentes significa desempenho similar em vídeos diferentes sem a necessidade de ajustes manuais ou qualquer outra intervenção humana [Han02].

Porém, grandes alterações na luminosidade da tomada ou casos extremos de movimentação de objetos e câmera são situações difíceis mesmo para um par característica/métrica robusto. Tais situações, combinadas às descontinuidades sutis nas transições graduais, requerem o uso de algum conhecimento extra para uma melhoria no desempenho do detector.

Várias informações extras podem auxiliar um detector. Conhecimento sobre os padrões assumidos por $z^l(i)$ durante um corte ou seu comportamento no decorrer de *fades* ou dissoluções são dados que podem ser utilizados na etapa de decisão para evitar falsas conclusões. No caso de dissoluções, por exemplo, Yeo e Liu [YL95] mostraram que, para l suficientemente grande, o gráfico de $z^l(i)$ assume a forma de um trapézio durante uma dissolução, o que os autores chamam de “platô”, como será visto mais adiante no Capítulo 3.

Além da análise de descontinuidades, há outras informações que podem auxiliar na detecção e na determinação da natureza de uma transição. Alattar [Ala93] observou que a variância da luminosidade dos quadros assume um padrão parabólico durante uma dissolução. Já Guimarães et al. [GCAL03] mostraram que há uma contração nos histogramas de luminosidade durante *fades*. Existem, também, certas informações *a priori* que podem ser utilizadas sem qualquer análise dos quadros da sequência de vídeo. Vasconcelos e Lippman [VL97, VL00] embutiram seu detector com informação sobre a *duração* das tomadas. Ime-

diatamente após o final de uma tomada, uma nova transição tem poucas chances de ocorrer. Contudo, a probabilidade de ocorrência de uma transição vai aumentando conforme o tempo passa. Tal idéia é retomada na abordagem probabilística de Hanjalic [Han02]. Algumas destas estratégias serão abordadas com mais detalhes no Capítulo 3.

2.5 Avaliação de desempenho

O sucesso de um delimitador de tomadas é medido pela acurácia com que ele identifica os extremos q_s e q_t de uma transição. As medidas de *reconhecimento* e *precisão*³, comumente utilizadas na área de recuperação de informação, são indicadores da qualidade da solução dada por um delimitador contra uma solução ideal anotada por um observador humano. Sua função é mensurar a quantidade de *falsos negativos* e *falsos positivos* em uma solução proposta, ou em outras palavras, avaliar a proporção de transições reais perdidas e falsas transições inseridas por engano do detector.

2.5.1 Avaliação para cortes

Considere $\mathcal{T}_{\text{corte}} = \{T_1, T_2, \dots, T_m\}$ o conjunto de cortes reais T_j existentes em uma seqüência de vídeo V (obtido por um avaliador humano ou através do esquema original de edição). Seja $\bar{\mathcal{T}}_{\text{corte}} = \{\bar{T}_1, \bar{T}_2, \dots, \bar{T}_p\}$ o conjunto de cortes $\bar{T}_i$ encontrados pelo delimitador para V . O reconhecimento $\mathcal{R}$, utilizado para avaliar a solução em relação a falsos negativos, é definido para cortes como

$$\mathcal{R}_{\text{corte}} = \frac{|\mathcal{T}_{\text{corte}} \cap \bar{\mathcal{T}}_{\text{corte}}|}{|\mathcal{T}_{\text{corte}}|}. \quad (2.4)$$

Para avaliar o número de falsos positivos da solução, utilizamos a precisão $\mathcal{P}$, definida por

³Os termos “reconhecimento” e “precisão” serão utilizados aqui no lugar das palavras inglesas “*recall*” e “*precision*” respectivamente.

$$\mathcal{P}_{\text{corte}} = \frac{|\mathcal{T}_{\text{corte}} \cap \bar{\mathcal{T}}_{\text{corte}}|}{|\bar{\mathcal{T}}_{\text{corte}}|}. \quad (2.5)$$

2.5.2 Avaliação para transições graduais

Transições graduais são também mais complexas com relação à avaliação de desempenho. A Figura 2.8 exibe duas situações que mostram que os critérios de reconhecimento e precisão utilizados anteriormente precisam ser redefinidos de modo a evitar que uma transição em $\bar{\mathcal{T}}_{\text{grad}}$, o conjunto das transições graduais $\bar{T}_i$ encontradas pelo detector, não seja associada a mais de uma transição em $\mathcal{T}_{\text{grad}}$, o conjunto das transições graduais T_j reais, e vice-versa. Como temos interesse também no reconhecimento do intervalo de quadros editados $q_s, \dots, q_t$, gostaríamos ainda de avaliar a solução proposta em relação aos limites encontrados para as transições.

Considere uma função *injetora* $f_{\text{grad}} : \bar{\mathcal{T}}_{\text{grad}} \rightarrow \mathcal{T}_{\text{grad}} \cup \{\Lambda\}$ tal que para cada $\bar{T}_i = (s_i, t_i)$ pertencente a $\bar{\mathcal{T}}_{\text{grad}}$

- i) $f_{\text{grad}}(\bar{T}_i) = T_j = (s_j, t_j)$ onde $[s_i, t_i] \cap [s_j, t_j] \neq \emptyset$ ou
- ii) $f_{\text{grad}}(\bar{T}_i) = \Lambda$.

Logo, uma transição proposta correta $\bar{T}_i$ deve possuir uma interseção com alguma transição T_j . O fato de f ser injetora impede que uma mesma $\bar{T}_i$ cubra várias transições reais, evitando assim soluções degeneradas como uma única transição $\bar{T} = (1, n)$.

Seja $\mathcal{T}_{\text{cov}} = \text{Img}(f_{\text{grad}}) \cap \mathcal{T}_{\text{grad}}$. Agora pode-se definir $\mathcal{R}_{\text{grad}}$ de modo similar ao visto anteriormente para cortes, como

$$\mathcal{R}_{\text{grad}} = \frac{|\mathcal{T}_{\text{cov}}|}{|\mathcal{T}_{\text{grad}}|}. \quad (2.6)$$

Analogamente, $\mathcal{P}_{\text{grad}}$ é definido por

$$\mathcal{P}_{\text{grad}} = \frac{|\mathcal{T}_{\text{cov}}|}{|\bar{\mathcal{T}}_{\text{grad}}|}. \quad (2.7)$$

Figura 2.8: Situação possível na avaliação do caso gradual. Uma transição proposta pode cobrir duas ou mais transições reais.

A proporção de quadros editados corretamente identificados por $\bar{T}_i$ é dada pela função $\text{rcov} : \bar{\mathcal{T}}_{\text{grad}} \rightarrow \mathbb{R}$, definida como

$$\text{rcov}(\bar{T}_i) = \begin{cases} \frac{|[s_i, t_i] \cap [s_j, t_j]|}{|[s_j, t_j]|} & \text{se } T_j = f_{\text{grad}}(\bar{T}) \neq \Lambda, \\ 0 & \text{caso contrário.} \end{cases} \quad \text{e} \quad (2.8)$$

De modo similar, a proporção de quadros erroneamente classificados pela mesma transição é dada pela função $\text{pcov} : \bar{\mathcal{T}}_{\text{grad}} \rightarrow \mathbb{R}$, dada por

$$\text{pcov}(\bar{T}_i) = \begin{cases} \frac{|[s_i, t_i] \cap [s_j, t_j]|}{|[s_i, t_i]|} & \text{se } T_j = f_{\text{grad}}(\bar{T}) \neq \Lambda, \\ 0 & \text{caso contrário.} \end{cases} \quad (2.9)$$

A cobertura de uma solução, isto é, as proporções totais de quadros correta e erroneamente identificados como quadros editados são definidas através dos valores $\mathcal{R}_{\text{cov}}$ e $\mathcal{P}_{\text{cov}}$, dados por

$$\mathcal{R}_{\text{cov}} = \frac{1}{|\mathcal{T}_{\text{cov}}|} \cdot \sum_{\bar{T}_i \in \bar{\mathcal{T}}_{\text{grad}}} \text{rcov}(\bar{T}_i) \quad \text{e} \quad (2.10)$$

$$\mathcal{P}_{\text{cov}} = \frac{1}{|\mathcal{T}_{\text{cov}}|} \cdot \sum_{\bar{T}_i \in \bar{\mathcal{T}}_{\text{grad}}} \text{pcov}(\bar{T}_i). \quad (2.11)$$

Capítulo 3

Revisão bibliográfica

Sendo um problema estudado há mais de dez anos, a delimitação de tomadas conta com várias propostas de soluções na literatura especializada. Idris e Panchanatan [IP97], Kopinska e Carrato [KC01] redigiram levantamentos sobre os algoritmos existentes. Boreczky e Rowe [BR96], Gargi *et al.* [GKS00] e Lienhart [Lie99, Lie01] apresentaram comparações entre alguns desses diversos métodos. O presente capítulo discorrerá sobre alguns dos métodos mais conhecidos da literatura, à luz do genérico Algoritmo 1 discutido no capítulo anterior, e apresentará resultados obtidos por alguns pesquisadores na comparação entre as várias propostas.

3.1 Características e métricas

Como visto na Seção 2.4, é comum utilizar uma medida de variação do conteúdo visual entre dois quadros como base para a detecção de transições. Diversas medidas foram propostas na literatura, baseadas em características espaciais (pixels, blocos e bordas), características de cor e na estrutura da compressão MPEG.

3.1.1 Domínio espacial: comparação pixel-a-pixel

A comparação direta entre os pixels correspondentes de dois quadros q_i e q_j foi uma das primeiras características exploradas na literatura. Uma das métricas mais simples é dada por

$$z_{\text{pixel}}(q_i, q_j) = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N |q_i[x, y] - q_j[x, y]|, \quad (3.1)$$

para quadros em níveis de cinza ou, de forma equivalente,

$$z_{\text{pixel}}(q_i, q_j) = \frac{1}{3MN} \sum_{c \in \{R, G, B\}} \sum_{x=1}^M \sum_{y=1}^N |q_i[x, y, c] - q_j[x, y, c]| \quad (3.2)$$

para quadros coloridos (no caso, no domínio RGB).

Infelizmente, tal métrica é incapaz de discernir entre uma pequena mudança em uma grande área dos quadros e uma grande alteração em uma pequena área, o que a torna muito sensível à movimentação de objetos e da câmera [IP97].

Devido a isso, Nagasaka e Tanaka [NT92] e Zhang *et al.* [ZKS93] propuseram considerar apenas os pixels onde a alteração de valor fosse considerável:

$$d_{\text{pixel}}(p_1, p_2) = \begin{cases} 1 & \text{se } |p_1 - p_2| > k_{\text{pixel}}, \\ 0 & \text{caso contrário.} \end{cases} \quad (3.3)$$

$$z_{\text{pixel}}(q_i, q_j) = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N d_{\text{pixel}}(q_i[x, y], q_j[x, y]), \quad (3.4)$$

onde p_1 e p_2 são valores em níveis de cinza (entre 0 e 255) e k_{pixel} um limiar fixado (a extensão para quadros coloridos decorre de forma análoga). Esta métrica é capaz de filtrar alterações irrelevantes entre pixels, mas ainda é muito sensível aos fatores extremos, sobretudo movimentação.

3.1.2 Domínio espacial: blocos

Buscando utilizar informação local para incrementar a robustez da métrica em situações de movimento, alguns autores substituíram a comparação entre pixels pela comparação de *blocos* de pixels, que dividem o quadro em regiões retangulares de mesmas dimensões e sem sobreposição entre si. Shahraray [Sha95] propôs a soma ponderada das diferenças entre blocos correspondentes:

$$z_{\text{bloco}}(q_i, q_j) = \sum_{b=1}^B c_b d_{\text{bloco}}(q_i[b], q_j[b]), \quad (3.5)$$

onde d_{bloco} mede a diferença entre dois blocos. Alguns autores [ZKS93, GKS00], assumindo $c_b = 1$ para todos os blocos, fazem uso da verossimilhança¹ proposta por Kasturi e Jain [KJ91]:

$$\lambda(q_i[b], q_j[b]) = \frac{\left[\frac{\sigma^2(q_i[b]) + \sigma^2(q_j[b])}{2} + \left(\frac{\mu(q_i[b]) - \mu(q_j[b])}{2} \right)^2 \right]^2}{\sigma^2(q_i[b]) \cdot \sigma^2(q_j[b])}, \quad (3.6)$$

$$d_{\text{bloco}}(q_i[b], q_j[b]) = \begin{cases} 1 & \text{se } \lambda(q_i[b], q_j[b]) > k_\lambda, \\ 0 & \text{caso contrário.} \end{cases} \quad (3.7)$$

onde $\sigma^2(q[b])$ e $\mu(q[b])$ são, respectivamente, a variância e a média da intensidade luminosa no bloco $q[b]$. Comparada à diferença pixel a pixel, esta métrica consegue reduzir o número de falsos positivos devido a sua tolerância a movimentos lentos de câmera e/ou objetos [ZKS93]. No entanto, pode gerar falsos negativos quando blocos muito diferentes têm probabilidades com funções de densidade similares, além do fato que os cálculos envolvidos nas equações 3.6 e 3.7 tornam muito lenta a computação da medida [ZKS93, XLI95, GKS00].

Xiong *et al.* [XLI95] estudaram como uma amostragem baseada em blocos poderia impactar a eficiência da segmentação. Eles argumentam que o tamanho do bloco tem estreita relação com a robustez do método a movimentações: quanto menor o bloco, mais ele se aproxima da comparação entre pixels e torna-se mais sensível a tais eventos. A partir dessa

¹Do original *likelihood*.

observação, os autores definem os blocos como quadrados de lado 2δ , onde δ é o máximo movimento, em pixels, entre duas imagens [BB82]. Eles alegam que, na prática, $\delta = 10$ é uma estimativa que leva a bons resultados. Isso garante que, não havendo uma transição abrupta, ao menos um quarto dos pixels de blocos correspondentes são iguais. A similaridade entre os blocos é mensurada através de

$$d_{\text{bloco}}(q_1[b], q_2[b]) = \begin{cases} 1 & \text{se } |\mu(q_1[b]) - \mu(q_2[b])| > k_{\text{bloco}}, \\ 0 & \text{caso contrário,} \end{cases} \quad (3.8)$$

onde $\mu(q[b])$ é a média da intensidade luminosa no bloco $q[b]$ e k_{bloco} um limiar (a equação 3.8 pode ser vista como generalização da equação 3.3). A diferença é dada por

$$z_{\text{bloco}}(q_i, q_j) = \frac{1}{|B_s|} \sum_{b \in B_s} d_b(q_i^b, q_j^b), \quad (3.9)$$

sendo B_s uma *amostra* do conjunto blocos B . Os autores provam que o desempenho da medida degenera muito pouco com a amostragem, podendo B_s ser consideravelmente menor que o conjunto de todos os blocos B .

3.1.3 Domínio espacial: bordas

Zabih *et al.* [ZMM95] propõem uma métrica baseada nos conceitos de *pixel de borda aferente*² e *pixel de borda eferente*³. Um pixel de borda aferente entre dois q_i e q_j é um pixel (x, y) de borda de q_j que não possua um correspondente em uma vizinhança de $q_i(x, y)$. Similarmente, um pixel de borda eferente é um pixel (x, y) de bordas de q_i que não possua um correspondente em uma vizinhança de $q_j(x, y)$.

Mais formalmente, seja f_i a imagem binária obtida por um detector de bordas (como o de Canny [Can86], por exemplo) aplicado a q_i . Seja também $\bar{f}_i$ a imagem binária obtida pela dilatação morfológica [GW00] de f_i por um elemento estruturante (um “diamante” de diagonais iguais a $2r + 1$ pixels). Define-se daí a *fração de pixels eferentes*, ρ_{out} , por

²Do original *entering edge pixel*.

³Do original *exiting edge pixel*.

$$\rho_{\text{out}}(q_i, q_j) = 1 - \frac{\sum_{x,y} f_i(x + \delta x, y + \delta y) \cdot \bar{f}_j(x, y)}{\sum_{x,y} f_i(x, y)}. \quad (3.10)$$

Analogamente, obtem-se a *fração de pixels aferentes*, ρ_{in} , por

$$\rho_{\text{in}}(q_i, q_j) = 1 - \frac{\sum_{x,y} \bar{f}_i(x + \delta x, y + \delta y) \cdot f_j(x, y)}{\sum_{x,y} f_i(x + \delta x, y + \delta y)}. \quad (3.11)$$

Os valores δx e δy são inteiros que representam uma translação que alinhe q_i e q_j , obtidos por algoritmos de compensação de movimento [BB95]. A partir daí, tem-se uma nova métrica, a *fração de bordas alteradas* z_ρ , dada por

$$z_\rho(q_i, q_j) = \max(\rho_{\text{out}}(q_i, q_j), \rho_{\text{in}}(q_i, q_j)). \quad (3.12)$$

A detecção de transições realizada pelos autores baseia-se nos padrões assumidos por ρ_{in} e ρ_{out} entre quadros sucessivos ao longo da seqüência de vídeo. Cortes apresentam picos em ρ_{in} e ρ_{out} . *Fades-in* apresentam valores crescentes de ρ_{in} ao longo do tempo, o mesmo ocorre de modo análogo com *fades-out* e ρ_{out} . Dissoluções são marcadas por valores crescentes de ρ_{out} e um pico de ρ_{in} e ρ_{out} na metade da transição, seguida por valores decrescentes de ρ_{in} .

3.1.4 Domínio de cores: histogramas

Uma das características mais utilizadas na literatura é o histograma de intensidade luminosa. O histograma de um quadro q_i em níveis de cinza no intervalo $[0, L - 1]$ é um vetor $h_i[l] = n_l$ onde l é o l -ésimo nível de cinza e n_l o número de pixels na imagem que apresentem tal valor [GW00]. Os elementos $h_i[l]$ são comumente chamados *bins*. Baseando-se nessa característica Nagasaka e Tanaka [NT92], Zhang *et al.* [ZKS93] propõem o uso da diferença entre histogramas *bin-a-bin*

$$z_h(q_i, q_j) = \sum_{l=0}^{L-1} |h_i[l] - h_j[l]|. \quad (3.13)$$

No caso de quadros coloridos, há várias alternativas. Uma delas é reduzi-los a quadros

em escala cinza. De acordo com o padrão de vídeo NTSC [IR98]:

$$l = 0,299 R + 0,587 G + 0,114 B. \quad (3.14)$$

Entretanto, existem várias formas para a construção de um histograma de cores propriamente dito (a Seção 3.5 apresentará algumas). Zhang *et al.*, por exemplo, utilizam um código para cada cor obtido tomando-se os dois *bits* mais significativos das componentes básicas R, G e B, como esquematizado na Figura 3.1.

Figura 3.1: Composição de um código de cor para o cômputo eficiente de histograma de cor segundo Zhang *et al.* [ZKS93].

Com esse código c , a variação por histograma pode ser calculada através de

$$z_h(q_i, q_j) = \sum_{c=0}^{63} |h_i[c] - h_j[c]|. \quad (3.15)$$

Há opções a diferença *bin-a-bin*. Nagasaka e Tanaka sugerem o uso do χ^2 como medida, revista posteriormente por Gargi *et al.* [GKS00]:

$$z_{\chi^2}(q_i, q_j) = \begin{cases} \sum_{l=1}^L \frac{(h_i[l] - h_j[l])^2}{h_j[l]} & \text{se } h_j[l] \neq 0, \\ \sum_{l=1}^L \frac{(h_i[l] - h_j[l])^2}{h_i[l]} & \text{caso contrário.} \end{cases} \quad (3.16)$$

Contudo, de acordo com os testes de Zhang *et al.* e Gargi *et al.*, o χ^2 não possui desempenho melhor que o de z_h 3.14, sendo ainda mais sensível às várias formas de movimentação. Por outro lado, Gargi *et al.* identificam em seus testes um melhor desempenho da medida dada pela *intersecção* de histogramas, definida por

$$z_{\cap}(q_i, q_j) = 1 - \frac{1}{MN} \sum_{l=1}^L \min(h_i[l], h_j[l]). \quad (3.17)$$

Quadros díspares podem ter histogramas similares, o que seria uma possível fonte de falsos negativos⁴. Histogramas são mais robustos à movimentação de objetos sobre um fundo constante do que as métricas em domínio espacial, mas permanecem sensíveis à movimentação de câmera [ZKS93, Han02].

3.2 Limiarização e detecção

A forma mais comum de delimitação de tomadas é utilizar o Algoritmo 1, exibido anteriormente no Capítulo 2, com $l = 1$ e $K(i) = k_{\text{corte}}$ para todo i . Em outras palavras, calcula-se a variação entre todo quadro q_i e seu sucessor imediato q_{i+1} , obtendo-se um sinal (Figura 3.2) que é posteriormente limiarizado frente a uma constante k_{corte} , um limiar global para todo o vídeo.

Figura 3.2: Gráfico $i \times z_h(q_i, q_{i+1})$ de um segmento de *Airline Safety and Economy*.

Embora seja uma solução razoável para a detecção de cortes, tal detector não tem um comportamento adequado com relação às transições graduais. A região entre os quadros 4500 e 4525, referente a uma dissolução em *Airline Safety and Economy*, ilustra a situação:

⁴A ocorrência de tais eventos é, felizmente, pouco freqüente.

um limiar global sobre o sinal poderia não identificar a transição com valores menores se comparados aos cortes em 4875, 4935 e 5028, ou ainda declarar como corte cada ponto da transição.

3.2.1 *Twin-comparison*

Transições graduais não podem ser detectadas analisando pontualmente $z(q_i, q_{i+1})$ [YL95]. Contudo, aplicando-se a medida de variação entre o primeiro e o último quadro de uma transição gradual, pode-se esperar um valor similar ao obtido em um corte.

Essa última observação é a base do método proposto por Zhang *et al.* [ZKS93] para detecção de cortes e transições graduais através de dois limiares: *twin-comparison*. O algoritmo descrito abaixo é uma implementação livre baseada na descrição do método pelos autores:

Algoritmo 2 (*Twin-comparison*) Dada uma seqüência de vídeo $V = \langle q_1, q_2, \dots, q_n \rangle$ e dois limiares k_{corte} e k_{grad} , $k_{\text{grad}} < k_{\text{corte}}$, encontra uma seqüência de transições $\mathcal{T} = \langle T_1, T_2, \dots, T_v \rangle$ em V .

- T1.** [Inicialize.] Atribua $i \leftarrow 1$, $s_{\text{grad}} \leftarrow \Lambda$ e $\mathcal{T} \leftarrow \langle \rangle$ (s_{grad} um possível início de uma transição gradual).
- T2.** [Calcule a variação.] Calcule $z(q_i, q_{i+1})$.
- T3.** [Corte?] Se $z(q_i, q_{i+1}) > k_{\text{corte}}$, adicione a nova a transição $T = (i, i + 1)$ a $\mathcal{T}$, atribua $s_{\text{grad}} \leftarrow \Lambda$ (pois não se trata de nenhuma transição gradual) e vá ao passo T6.
- T4.** [Início de transição gradual?] Se $z(q_i, q_{i+1}) > k_{\text{grad}}$ e $s_{\text{grad}} = \Lambda$, atribua $s_{\text{grad}} \leftarrow i$ e vá ao passo T6.
- T5.** [Fim de transição gradual?] Se $z(q_i, q_{i+1}) \leq k_{\text{grad}}$, $s_{\text{grad}} \neq \Lambda$ e $z(s_{\text{grad}}, i) > k_{\text{corte}}$ e adicione a nova transição $T = (s_{\text{grad}}, i)$ a $\mathcal{T}$. Atribua $s_{\text{grad}} \leftarrow \Lambda$.
- T6.** [Incremente i .] Atribua $i \leftarrow i + 1$.

T7. [Fim do vídeo?] Se $i = n$, o algoritmo termina. Caso contrário, volte a T2. ■

No algoritmo acima, k_{grad} é utilizado para avaliar possíveis limites de uma transição gradual. Quando $z(q_i, q_{i+1}) > k_{\text{grad}}$, toma-se um possível início para uma transição gradual. No momento em que $z(q_i, q_{i+1}) \leq k_{\text{grad}}$, é avaliado se a diferença acumulada no período excede k_{corte} , um limiar que indica uma aparente troca de contexto (o mesmo utilizado para cortes). *Twin-comparison* é um algoritmo simples com bons resultados e que pode ser utilizado com as várias métricas definidas na Seção 3.1.

3.2.2 Escala temporal e “platôs” de variação

Se a análise pontual $z(q_i, q_{i+l})$ com $l = 1$, entre q_i e seu sucessor, não é suficiente para identificar transições graduais, o mesmo não vale para outros valores de l . Yeo e Liu [YL95] mostram que, se l é maior que a duração de uma transição gradual, é possível identificá-la. Em tal condição, o sinal dado por $z^l(i)$ tomará uma forma muito característica durante as transições graduais: um *platô*.

Para entender o método, é necessário pensar novamente em um modelo linear para as transições graduais. Considere uma função $g(i)$ com uma mudança linear de um valor c_1 para outro valor c_2 entre dois pontos s e t :

$$g(i) = \begin{cases} c_1 & i < s, \\ \frac{c_2 - c_1}{t - s}(i - s) + c_1 & s \leq i < t, \\ c_2 & i \geq t. \end{cases} \quad (3.18)$$

Assuma $z^l(i) = g(i + l) - g(i)$. Observando a Figura 3.3, pode-se ver que, se $l > t - s$, tem-se

$$z^l(i) = \begin{cases} 0 & i < s-l, \\ \frac{|c_2-c_1|}{t-s}[i-(s-l)] & s-l \leq i < t-l, \\ |c_2-c_1| & t-l \leq i < s, \\ -\frac{|c_2-c_1|}{t-s}(i-t) & s \leq i < t, \\ 0 & i \geq t. \end{cases} \quad (3.19)$$

Conclui-se, então, que surge um “platô” constante ($|c_2 - c_1|$) imediatamente antes da transição. Tal suposição encontra forte evidência empírica, como pode ser visto na Figura 3.4, onde exibe-se $z_h(q_i, q_{i+30})$.

Figura 3.3: Os cinco intervalos diferentes na Equação 3.19.

Yeo e Liu buscam transições graduais dentro de um sinal $z^l(i)$ procurando por regiões que atendam os requisitos

1. $z^l(i) \approx z^l(j)$, $j = i - p, \dots, i - 1, i + 1, \dots, i + p$ e
2. $z^l(i) \geq k_r \cdot z^l(i - \lfloor l/2 \rfloor - 1)$ e $z^l(i) \geq k_r \cdot z^l(i + \lfloor l/2 \rfloor + 1)$, para duas constantes k_r e p definidas.

As duas condições destinam-se a identificar a ocorrência dos “platôs”. É importante notar que o método funciona mesmo no caso das transições não serem lineares (como o caso da função α descrita na Seção 2.2.1), bastando apenas a existência do “platô”, independentemente da forma como ele “ascende” ou “descende”.

Figura 3.4: Gráfico $i \times z_h(q_i, q_{i+30})$ de um segmento de *Airline Safety and Economy*.

3.2.3 Limiares adaptativos

Como apresentado anteriormente no Capítulo 2, pode haver uma intersecção entre os intervalos de valores assumidos por diferenças intra-tomada e diferenças inter-tomada e uma função $K(i)$ adequada é necessária para melhorar o desempenho do detector. Como pode ser visto na Figura 3.5, referente a um trecho de *The Corvair in Action*, não há limiar global capaz de detectar corretamente todas as transições sem gerar muitos falsos positivos.

Yeo e Liu [YL95] propuseram o uso de um limiar adaptativo calculado sobre uma *janela deslizante*⁵. Para eles, um detector de cortes deve buscar por quadros i tais que $z^1(i) > K(i)$, onde $K(i)$ é dado por

$$K(i) = c \cdot \max(z^1(i - m + 1), \dots, z^1(i - 1), z^1(i + 1), \dots, z^1(i + m - 1)), \quad (3.20)$$

ou, em outras palavras, $z^1(i)$ deve ser não só o máximo dentro de uma janela de tamanho

⁵Do original *sliding window*.

Figura 3.5: Gráfico $i \times z_h(q_i, q_{i+1})$ de um segmento de *The Corvair in Action*.

$2m - 1$ como também c vezes maior que a segunda maior variação dentro da mesma janela, onde c é uma constante definida (os autores relatam que valores entre 2 e 3 geram melhores resultados). O parâmetro m deve ser menor que a duração mínima de uma tomada para evitar falsos negativos. Assumindo que uma tomada deveria durar ao menos 1 segundo, $m = 30$ seria uma escolha razoável para segmentar um vídeo digital a 30 quadros/segundo.

Gargi *et al.* [GKS00], ao comparar diversas características e métricas, utilizaram limiares adaptativos em seus testes. Os autores utilizaram como limiar $K(i)$ a média das variações em uma janela de tamanho 12 centrada em i , multiplicada por uma constante c . Embora não informem maiores detalhes sobre a constante utilizada, Gargi *et al.* relatam que esta forma de limiarização adaptativa se mostrou efetiva.

3.2.4 Step-variable

Como visto anteriormente com *twin-comparison* (Seção 3.2.1) e no método de Yeo e Liu (Seção 3.2.2), a detecção de transições graduais pode ser feita comparando-se quadros pertencentes aos extremos da transição. Com este objetivo e ainda o de diminuir o tempo total de detecção, Xiong e Lee [XL98] desenvolveram um algoritmo chamado *step-variable*:

Algoritmo 3 (*Step-variable*) Dada uma seqüência de vídeo $V = \langle q_1, q_2, \dots, q_n \rangle$ e três valores k_b , k_e e l , encontra uma seqüência de transições $\mathcal{T} = \langle T_1, T_2, \dots, T_v \rangle$ em V .

- S1.** [Inicialize.] Atribua $i \leftarrow 1$ e $\mathcal{T} \leftarrow \langle \rangle$. Atribua $j \leftarrow i + l$.
- S2.** [Calcule a variação.] Calcule $z(q_i, q_j)$.
- S3.** [Excedeu?] Se $z(q_i, q_j) \leq k_b$ siga para S8. Caso contrário, atribua $i_0 \leftarrow i$ e $j_0 \leftarrow j$.
- S4.** [Busca binária] Se $j = i + 1$, vá para S5. Caso contrário, vá para S7.
- S5.** [Corte?] Se $z(q_i, q_j) > k_b$, adicione a nova transição $T = (i, j)$ a $\mathcal{T}$ e vá ao passo S8. Caso contrário, prossiga a S6.
- S6.** [Transição gradual?] Se $z_e(q_{i_0}, q_{j_0}) > k_e$, adicione a nova transição $T = (i_0, j_0)$ a $\mathcal{T}$, atribua $i \leftarrow j + 1$, $j \leftarrow i + l$ e volte a S2. Caso contrário, siga para S8.
- S7.** [Redução] Atribua $m \leftarrow \lfloor \frac{i+j}{2} \rfloor$. Calcule $z(q_i, q_m)$ e $z(q_m, q_j)$. Se $z(q_i, q_m) > z(q_m, q_j)$, atribua $i \leftarrow m$. Caso contrário, atribua $j \leftarrow m$. Volte a S4.
- S8.** [Incremente i e j .] Incremente $i \leftarrow i + \frac{l}{2}$. Atribua agora $j \leftarrow i + l$.
- S9.** [Fim do vídeo?] Se $j > n$, o algoritmo termina. Caso contrário, volte a S2. ■

A detecção de transições graduais, no passo S6, utiliza uma medida $z_e(q_i, q_j)$ baseada na diferença entre as bordas dos dois quadros. Entretanto, os autores não entram em detalhes quanto a sua implementação.

Considere $\max_{\text{grad}}$ o tamanho máximo assumido por uma transição gradual no vídeo V e $\min_S$ a menor duração de uma tomada em V . Se

$$2 \cdot \max_{\text{grad}} < l < \min_S \quad (3.21)$$

o Algoritmo 3 terá a capacidade de isolar todo corte ou transição gradual existente em V entre dois quadros q_i e q_j tais que q_i se encontre à esquerda da transição e q_j à direita.

Teorema 1 (Xiong e Lee 1998 [XL98]) *Se $2 \cdot \max_g < l < \min_S$, o Algoritmo 3 sempre obterá, para toda transição $T = (s, t)$, um par de índices i e j tais que $i \leq s$ e $j \geq t$, não havendo mais do que uma transição entre esse par de quadros.*

Prova: Dado que $l < \min_S$, para toda tomada $S \in V$ haverá um valor i tal que $q_i \in S$, de forma que sempre haverá um q_i à esquerda de cada transição. Assim, há sempre três situações possíveis:

Situação 1 Ambos q_i e q_j se encontram à esquerda de uma transição, como na Figura 3.6 (a).

O próximo valor de i, i' , não atingirá ou passará da transição, pois $i < i' < j$. Logo, o novo par i' e j' deverá obedecer uma das duas próximas situações.

Situação 2 O quadro q_i está à esquerda da transição enquanto que q_j está no meio de uma transição gradual, ilustrado na Figura 3.6 (b). Como $2 \cdot \max_{\text{grad}} > l$, $i' = i + \frac{l}{2}$ não atingirá ou passará da transição, mas $j' = i' + l$ se encontrará à direita dela, o que leva ao último caso:

Situação 3 O quadro q_i se encontra à esquerda da transição e q_j à direita. É a situação desejada, vista na Figura 3.6 (c). ■

Dessa forma, o passo S3 sempre poderá isolar todas as transições, cabendo ao teste de variação detectá-la. Como visto no passo S6, o algoritmo consegue detectar a ocorrência da uma transição gradual, mas não é capaz de delimitar sua duração. Os autores propõem uma variação do algoritmo para encontrar os limites de uma transição gradual, incrementando i e decrementando j até atingir os extremos, mas sua descrição não é clara.

3.3 Informação adicional sobre os quadros

Alattar [Ala93], ao buscar um detector de dissoluções para incrementar o desempenho de algoritmos de compensação de movimento da Intel Corporation, mostrou que a variância da luminosidade do quadro $\sigma^2(q_i)$ assume, ao longo do tempo, uma forma parabólica.

Considerando o modelo de dissolução baseado em uma função $\alpha(j) = \frac{j-s}{t-s}$ e assumindo que a luminosidade nos quadros de $S_i = \langle a_1, a_2, \dots, a_m \rangle$ é estatisticamente independente da luminosidade nos quadros de $S_{i+1} = \langle b_1, b_2, \dots, b_n \rangle$ temos que a variância da luminosidade neste trecho do vídeo V é dada por

Figura 3.6: Os incrementos do algoritmo *Step-variable* de Xiong e Lee [XL98].

$$\sigma_V^2(j) = \begin{cases} \sigma_i^2(j) & j < s, \\ (1 - \alpha(j))^2 \cdot \sigma_i^2(j) + \alpha^2(j) \cdot \sigma_{i+1}^2(j) & s \leq j \leq t, \\ \sigma_{i+1}^2(j) & j > t. \end{cases} \quad (3.22)$$

Assumindo que a sequência de quadros de S_i e S_{i+1} formam processos aleatórios e ergódicos, as variâncias $\sigma_i^2(j)$ e $\sigma_{i+1}^2(j)$ tornam-se constantes e a Equação 3.22 pode ser reduzida para

$$\sigma_V^2(j) = \begin{cases} \sigma_i^2 & j < s, \\ (\sigma_i^2 + \sigma_{i+1}^2) \cdot \alpha^2(j) - 2\sigma_i^2 \cdot \alpha(j) + \sigma_i^2 & s \leq j \leq t, \\ \sigma_{i+1}^2 & j > t. \end{cases} \quad (3.23)$$

Fica claro, na Equação 3.23, que a variância do vídeo assume uma forma parabólica, com concavidade para cima, no trecho entre s e t . Tomando-se a segunda derivada de $\sigma_V^2(j)$ em relação a j :

$$\frac{d^2 \sigma_V^2(j)}{dj} = \begin{cases} 0 & j < s, \\ \frac{2}{(t-s)^2} (\sigma_i^2 + \sigma_{i+1}^2) & s \leq j \leq t, \\ 0 & j > t. \end{cases} \quad (3.24)$$

A Equação 3.26 indica que uma dissolução pode ser detectada buscando-se trechos em V com um valor constante diferente de zero em $\frac{d^2 \sigma_V^2(j)}{dj}$. Contudo, essa equação é válida somente para o caso contínuo. O vídeo V é discreto e j encontra-se nos naturais $\mathbb{N}$. Nesse caso, pode-se utilizar uma diferença discreta $d''f$ dada por

$$d''f(j) = f(j) - 2f(j-1) + f(j-2), \quad (3.25)$$

que leva, em relação a $\sigma_V^2(j)$, a

$$d''\sigma_V^2(j) = \begin{cases} 0 & j \leq s, \\ \frac{1}{(t-s)^2}(\sigma_i^2 + \sigma_{i+1}^2) - \frac{2}{t-s}\sigma_i^2 & j = s+1 \\ \frac{2}{(t-s)^2}(\sigma_i^2 + \sigma_{i+1}^2) & s+1 < j \leq t, \\ \frac{1}{(t-s)^2}(\sigma_i^2 + \sigma_{i+1}^2) - \frac{2}{t-s}\sigma_{i+1}^2 & j = t+1 \\ 0 & j > t+1. \end{cases} \quad (3.26)$$

O caso discreto leva a dois picos negativos nas posições $s+1$ e $t+1$, pontos que marcam o início e o fim da parábola, e, conseqüentemente, às extremidades da dissolução. Tal modelo teórico da transição encontra forte evidência na prática, como visto na Figura 3.7, onde há três dissoluções centradas na altura dos quadros 6980, 7150 e 7480.

Figura 3.7: Gráfico $i \times \sigma^2(q_i)$ de um segmento de *Airline Safety and Economy* contendo três dissoluções.

Truong *et al.* [TDV00] partem do trabalho de Alattar para desenvolver algoritmos mais sofisticados à detecção e delimitação de *fades* e dissoluções. Os autores observam que, durante um *fade*, a média da luminosidade dos quadros move-se linearmente até atingir um valor constante, como pode ser visto na Figura 3.8, onde o vídeo se inicia com um *fade-in* até o quadro 23. Em seguida, tem-se um *fade-out* do quadro 175 ao 200, seguido por um novo *fade-in* entre os quadros 250 e 275. Paralelamente, a variância da luminosidade assume

a forma de meia parábola, como exibido na Figura 3.9 para o mesmo trecho.

A partir daí, os autores definem seu método para a detecção de *fades*. Inicialmente, procuram por quadros monocromáticos ($\sigma_V^2(j) \approx 0$), típicos no início de *fades-in* e no final de *fades-out*. De modo similar a Alattar, buscam por picos na segunda derivada de $\sigma^2(q_i)$ próximos aos quadros monocromáticos, possíveis extremidades para a transição. Finalmente, estudam o comportamento da primeira derivada da média da luminosidade, que deveria ser constante durante o *fade*.

Figura 3.8: Gráfico $i \times \mu(q_i)$ de um segmento de *Airline Safety and Economy* contendo *fades*.

Figura 3.9: Gráfico $i \times \sigma^2(q_i)$ de um segmento de *Airline Safety and Economy* contendo *fades*.

Na detecção de dissoluções, Truong *et al.* alegam que o método de Alattar é suscetível

aos efeitos do ruído causado pela movimentação de câmera e objetos, que pode gerar também picos negativos na segunda derivada da variância. Os autores refinam o método de Alattar buscando, na primeira derivada do sinal das variâncias, por cruzamentos em zero, como visto na Figura 3.10. O sinal da derivada passa posteriormente por uma filtragem passa-baixas [GW00] para eliminação de ruídos. Algebricamente, os autores definem restrições (na forma de limiares) para a diferença entre a variância nas extremidades e no meio da dissolução, como forma de assegurar o caráter parabólico do sinal.

Figura 3.10: Gráfico $i \times \frac{d\sigma^2(q_i)}{di}$ de um segmento de *Airline Safety and Economy* contendo três dissoluções.

3.4 Filtragem

O sinal dado por $z^l(i)$, independente da métrica, é sujeito a ruídos, decorrentes principalmente da movimentação de objetos e câmera. O uso de limiares adaptativos é uma forma de tratar o problema. Uma outra alternativa, explorada por Ferman e Tekalp [FT98] é a filtragem do sinal. O objetivo é eliminar ruídos intra-tomadas, mas preservando os picos inter-tomadas. O problema é visto de modo similar à remoção de ruídos em processamento de imagens: deseja-se eliminar ruído (movimentação intra-tomada) sem, contudo, deteriorar as bordas da imagem (transição inter-tomada).

Considere $\tilde{z}_l(i)$ como o sinal obtido por uma filtragem passa-baixas sobre $z^l(i)$. Ferman

e Tekalp propõem o uso, na detecção, do sinal $z'_l(i)$ dado por:

$$z'_l(i) = \begin{cases} z^l(i) - \tilde{z}_l(i) & \text{se } z^l(i) - \tilde{z}_l(i) > 0, \\ 0 & \text{caso contrário.} \end{cases} \quad (3.27)$$

Os autores fazem uso da média e da mediana local, dentro de uma janela, para a obtenção de $\tilde{z}_l(i)$. O tamanho da janela é escolhido de forma adaptativa: as janelas que possuem maior variância em $z^l(i)$ são utilizadas no cálculo da média, como forma de eliminar ruído decorrente de movimentação, enquanto que janelas com menor variância são utilizadas no cálculo de medianas.

Ferman e Tekalp relatam que melhores resultados podem ser obtidos com a obtenção de $z'_l(i)$ pela subtração das médias seguida de uma segunda filtragem, obtendo $z''_l(i)$ pela subtração das medianas.

3.5 Comparação entre métodos

Alguns pesquisadores dedicaram trabalhos à comparação entre as várias técnicas de segmentação temporal, enfatizando principalmente os resultados de reconhecimento e precisão descritos anteriormente na Seção 2.5.

Boreczky e Rowe [BR96] compararam dois algoritmos baseados no sinal da variação por histogramas: limiar global aplicado ao sinal e *twin-comparison*. Um terceiro algoritmo, baseado na diferença entre blocos obtida via histogramas também foi testado. Completavam os testes um algoritmo baseado em compensação de movimento e outro baseado em coeficientes DCT⁶. Os resultados reportados por Boreczky e Rowe indicam um melhor desempenho dos algoritmos baseados em histogramas. O algoritmo baseado em coeficientes DCT teve desempenho inferior aos demais.

Gargi *et al.* [GKS00] compararam os efeitos das várias representações de cor no desempenho das medidas z_h , z_{χ^2} e $z_{\cap}$. Testes foram realizados nos domínios RGB, HSV, YIQ,

⁶De *discrete cosine transform* [GW00], transformada utilizada na compressão de vídeo no padrão MPEG.

XYZ, $L^*a^*b^*$, $L^*u^*v^*$, Munsell e eixos oponentes (OCA⁷), além da simples escala cinza, para analisar a importância da informação de cor na delimitação, sendo os histogramas representados de três formas diferentes: 1D, 2D e 3D. Na representação 1D, cada componente é representada por 256 *bins*, formando um vetor de 768 valores. Na representação 2D, uma matriz bidimensional 16×16 foi utilizada, representando em cada posição um par de valores para cromaticidade, descartando a informação de luminância, ou seja, H e S em HSV, a^* e b^* para $L^*a^*b^*$ e assim sucessivamente. Já a representação 3D utilizou uma matriz tridimensional $8 \times 8 \times 8$ contendo a contagem de ocorrências de cada trinca de valores. Com relação à quantificação, cada componente pôde assumir 256 valores possíveis na representação 1D, 16 na 2D e 8 na 3D. Histogramas no domínio RGB não têm representação 2D.

Os resultados obtidos indicam um melhor desempenho do espaço de Munsell, seguido por $L^*a^*b^*$, $L^*u^*v^*$ e OCA. Como dito anteriormente, a interseção de histogramas tem melhor desempenho que as diferenças *bin-a-bin* ou o χ^2 . Representações 1D e 3D tiveram desempenho similar enquanto que a representação 2D proposta apresentou resultados mais pobres, indicando a importância da informação de luminosidade na medida.

⁷De *opponent color axis*.

Capítulo 4

Refinamentos

4.1 Distribuição das diferenças entre quadros adjacentes

A Seção 3.2 apresentou uma forma simples de detecção de cortes: comparar o sinal de similaridade a um limiar k_{corte} . Mas, como visto na Seção 2.4, há sobreposição dos intervalos de valores assumidos pela diferença entre quadros dentro de uma mesma tomada ($\bar{R}$) e entre quadros em tomadas diferentes (R), o que compromete o desempenho do método.

A Figura 4.1 ilustra o problema. Ela se refere à distribuição de $z_{\cap}^1(i)$ em um vídeo com cerca de 31 minutos (um total de 56716 quadros) obtido do canal CNN em 03 de fevereiro de 1998 pelo Linguistic Data Consortium (LDC) e disponibilizado ao NIST para o TRECVID'04. Todas as transições foram anotadas pelo LDC/NIST e fornecidas através de um arquivo XML. No decorrer deste capítulo este vídeo será referenciado como *CNN 03/02/1998*.

Certamente, não há um limiar global k_{corte} capaz de classificar corretamente todo quadro de CNN 03/02/1998 como pertencente a uma transição (limite de tomada) ou como um quadro interno, pois há várias intersecções entre as distribuições de $z_{\cap}^1(i)$ em cortes, transições graduais e entre quadros de uma mesma tomada. Haveria contudo um limiar capaz de atingir

Figura 4.1: Distribuição de $z_{\cap}(q_i, q_{i+1})$ em CNN 03/02/1998.

altos níveis de acerto?

Zhang *et al.* [ZKS93] assumem que a distribuição das variações intra-tomadas aproxime-se de uma gaussiana $N(\mu_z, \sigma_z)$. A partir daí, os autores propõem o uso de limiares da forma

$$k_{\text{corte}} = \mu_z + \alpha \sigma_z. \quad (4.1)$$

De acordo com essa hipótese, um valor $\alpha = 3$ seria capaz de agrupar cerca de 99,9% de todos os valores intra-tomadas. Infelizmente, aplicando-se o Algoritmo 2, não se consegue tal desempenho. A Tabela 4.1 exibe o resultado obtido na detecção de cortes em CNN 03/02/1998:

α	Reconhecimento	Precisão
3	0.8321	0.2442
4	0.6928	0.2721
5	0.5286	0.2686
6	0.4036	0.2846

Tabela 4.1: Resultados na detecção de cortes obtidos por limiares globais aplicados sobre $z_{\cap}^1(i)$ e calculados segundo a Equação 4.1 em CNN 03/02/1998.

O desempenho pobre dos limiares selecionados segundo a Equação 4.1 pode ser compreendido analisando-se o histograma da distribuição dos valores de diferenças intra-tomadas. É possível notar que a distribuição normal não é uma boa aproximação para a distribuição dessas diferenças. Hanjalic [Han02] afirma que melhores estimativas podem ser encontradas na família de funções

$$p(z(i)|\bar{R}) = h_1 z^1(i)^{h_2} e^{-h_3 z^1(i)}, \quad (4.2)$$

embora acredite que a distribuição das diferenças em cortes siga realmente a distribuição normal (como pode ser intuído na Figura 4.1).

4.2 Análise local

Analisar o sinal localmente pode levar a resultados superiores. Na Seção 3.2 viu-se que Yeo e Liu [YL95] buscam a diferença máxima em uma janela e a comparam com o segundo máximo no mesmo local de modo a se certificarem que o valor destoa consideravelmente do comportamento médio assumido pelo sinal das diferenças naquele trecho do vídeo. Hanjalic [Han02] e outros propõem uma pequena variação ao método de Yeo e Liu: ao invés de utilizar a razão entre os dois maiores valores na janela, calcular a *distância relativa* entre os dois valores, dada por

$$\psi^m(i) = \begin{cases} 100 \cdot \frac{z^1(i) - \text{smax}(i, m)}{z^1(i)} \% & \text{se } z^1(i) \geq z^1(j), \forall j \in [i - m, i + m], \\ 0 & \text{caso contrário.} \end{cases} \quad (4.3)$$

onde $\text{smax}(i, m)$ é o valor máximo encontrado em $z^1(i - m), \dots, z^1(i - 1), z^1(i + 1), \dots, z^1(i + m)$. Sendo $z^1(i)$ o máximo em uma janela de tamanho $2m + 1$, centrada em i , $\text{smax}(i, m)$ será o segundo maior valor nessa janela e iremos avaliar a distância relativa entre esses dois valores. A Tabela 4.2 exhibe os resultados obtidos pela aplicação de vários limiares ao sinal $\psi_{\cap}^7(i)$ ¹.

k_c	Reconhecimento	Precisão
50%	0.7464	0.8745
60%	0.6678	0.9350
70%	0.5500	0.9448
80%	0.3464	0.9327

Tabela 4.2: Resultados na detecção de cortes obtidos por limiares globais sobre $\psi_{\cap}^7(i)$ em CNN 03/02/1998.

Vários filtros podem ser desenvolvidos seguindo a mesma idéia de favorecer a análise local. Truong *et al.* [TDV00] utilizam a *razão pela média* em uma janela de tamanho

¹A distância relativa obtida para $z_{\cap}^1(i)$.

Figura 4.2: Distribuição de $\psi_{\cap}^7(i)$ em CNN 03/02/1998.

Figura 4.3: *Boxplots* de $z_{\cap}^1(i)$ e suas várias filtragens, referente a CNN 03/02/1998. Os quartis permitem avaliar a grande interseção entre as distribuições inter e intra-tomadas.

$2m + 1$, excetuando-se porém o valor central $z^1(i)$:

$$\varrho^m(i) = \frac{z^1(i) + c}{\frac{\sum_{j=i-m, j \neq i}^{i+m} (z^1(j) + c)}{2m}}, \quad (4.4)$$

onde c é uma constante utilizada para tratar casos onde os valores $z^1(j)$ são zero, evento comum decorrente de uma seqüência de dois ou mais quadros estáticos. Outros filtros, apresentados anteriormente na Seção 3.4, são utilizados por Ferman e Tekalp e [FT98], baseiam-se na *subtração da média ou mediana*. A Figura 4.3 exhibe o efeito obtido pelas filtragens na distribuição das várias categorias de transições (e sua capacidade em diminuir a interseção entre elas). É importante notar que os filtros apresentados aqui são alternativas à aplicação de limiares adaptativos, pois as fórmulas apresentadas podem ser utilizadas na criação de funções $K(i)$ [TDV00]. A Figura 4.4 compara o desempenho obtido pelos 3 filtros aplicados sobre $z_{\cap}^1(i)$.

4.2.1 Reconhecimento: perda de cortes

Um dos maiores problemas da delimitação de tomadas em vídeos de várias origens e naturezas (noticiários, comerciais, entrevistas, filmes,...) são os diversos eventos que podem ocorrer. Ao utilizar $\psi_{\cap}^7$ um detector busca pela diferença que tenha o valor máximo em uma janela de tamanho $2m + 1 = 15$ e que seja também consideravelmente superior ao segundo maior valor dessa janela. A Figura 4.5 mostra um trecho de CNN 03/02/1998 no qual é exibida uma chamada comercial da série de ficção científica *Babylon 5*. No processo de edição, foram criadas tomadas com apenas 4 ou 3 quadros de duração. Infelizmente, os cortes nos quadros 56726, 56730 e 56733 se anulam na Equação 4.3 e nenhum dos três é detectado. Partindo-se da premissa que uma tomada deve ter ao menos 8 quadros (aproximadamente um quarto de segundo), obtém-se os resultados exibidos na Tabela 4.3.

No entanto, a principal fonte de falsos negativos é a limitação do histograma de cor como característica discriminante. Há tomadas cujas distribuições de cores são muito próximas, como na Figura 4.6 (b), não havendo uma resposta suficientemente alta no sinal $z^1(i)$. Essas

Figura 4.4: Comparação entre os desempenhos na detecção de cortes (reconhecimento $\times$ precisão) obtidos pela aplicação de quatro filtros diferentes sobre $z_{\cap}^1(i)$.

k_c	Reconhecimento	Precisão
10%	0.9540	0.2299
20%	0.9310	0.4764
30%	0.9119	0.6629
40%	0.8889	0.8227
50%	0.8506	0.9289
60%	0.7509	0.9800

Tabela 4.3: Resultados na detecção de cortes obtidos por limiares globais sobre $\psi_{\cap}^7(i)$ em CNN 03/02/1998 após a remoção manual de tomadas com comprimento menor do que 8 quadros.

Figura 4.5: Uma seqüência de ação em uma chamada comercial da série de ficção científica *Babylon 5* em CNN 03/02/1998. Os cortes que delimitam as tomadas curtas anulam-se em $\psi_{\cap}^7(i)$.

Figura 4.6: Falsos-negativos em CNN 03/02/1998, perdidos tomando-se $\psi_{\cap}^7(i) > 50\%$. Quadros escuros costumam levar a erros (a). Da mesma forma, distribuições de cores similares (b) ou com cenário comum (c) geram valores muito baixos em diferenças baseadas em histogramas.

situações são comuns sobretudo em casos onde o mesmo cenário ou fundo é utilizado em duas tomadas sucessivas, sem alteração de ângulo da câmera, como pode ser visto na Figura 4.6 (c), ou em situações onde a luminosidade é baixa, como na Figura 4.6 (a).

4.3 Transições graduais

Como visto anteriormente, o Algoritmo 2 (*twin-comparison*) é um algoritmo capaz de detectar transições graduais. Contudo, se os limiares utilizados forem os mesmos para todos os trechos do vídeo, o desempenho geralmente será baixo.

Para obter melhores resultados, o sinal $\psi^m(i)$ poderia ser utilizado de maneira similar. Porém, como pode ser visto comparando as Figuras 4.1 e 4.2, o filtro definido na Equação 4.3 tem um efeito indesejado em transições graduais, concentrando os valores das diferenças durante a transição próximo de zero, dificultando sua separação das diferenças intra-tomadas.

Propõem-se aqui uma variação do Algoritmo 2, que utiliza tanto o sinal filtrado como o sinal original. Ela se baseia novamente na diferença acumulada entre os quadros inicial e final da transição gradual, q_s e q_t . A idéia é simples: ao surgir um trecho considerado uma possível transição gradual, ele é removido da seqüência, deixando-se apenas seus extremos q_s e q_t . Então, $\psi^m(s)$ é recalculado, e o resultado comparado a k_{corte} . Em outras palavras, reduzimos a transição gradual a um corte, aplicamos a filtragem e comparamos o resultado ao limiar definido.

Algoritmo 4 (*Twin-comparison* ψ) Dada uma seqüência de vídeo $V = \langle q_1, q_2, \dots, q_n \rangle$, dois limiares k_{corte}^ψ e k_{grad} e um valor m para filtragem local, encontra uma seqüência de transições $\mathcal{T} = \langle T_1, T_2, \dots, T_v \rangle$ em V .

- T1.** [Inicialize.] Atribua $i \leftarrow 1$, $s_{\text{grad}} \leftarrow \Lambda$ e $\mathcal{T} \leftarrow \langle \rangle$ (s_{grad} um possível início de uma transição gradual).
- T2.** [Calcule a variação.] Calcule $z^1(i) = z(q_i, q_{i+1})$ para todo $1 \leq i < n$. Obtenha então $\psi^m(i)$.
- T3.** [Corte?] Se $\psi^m(i) > k_{\text{corte}}^\psi$, adicione a nova a transição $T = (i, i + 1)$ a $\mathcal{T}$, atribua $s_{\text{grad}} \leftarrow \Lambda$ (pois não se trata de nenhuma transição gradual) e vá ao passo T6.
- T4.** [Início de transição gradual?] Se $z^1(i) > k_{\text{grad}}$ e $s_{\text{grad}} = \Lambda$, atribua $s_{\text{grad}} \leftarrow i$ e vá ao passo T6.
- T5.** [Fim de transição gradual?] Se $z^1(i) \leq k_{\text{grad}}$ e $s_{\text{grad}} \neq \Lambda$, calcule $\psi^m(s_{\text{grad}})$ para a seqüência $\langle s_{\text{grad}} - m, s_{\text{grad}} - m + 1, \dots, s_{\text{grad}}, i, i + 1, i + m \rangle$. Daí, se $\psi^m(s_{\text{grad}}) > k_{\text{corte}}^\psi$, adicione a nova transição $T = (s_{\text{grad}}, i)$ a $\mathcal{T}$. Atribua $s_{\text{grad}} \leftarrow \Lambda$.

k_{corte}	k_{grad}	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}_{\text{cov}}$	$\mathcal{P}_{\text{cov}}$
40%	0.05	0.8947	0.3796	0.7096	0.7182
	0.06	0.8538	0.5123	0.6384	0.8295
	0.07	0.7778	0.5733	0.5931	0.8471
	0.08	0.6608	0.5855	0.5502	0.9024
	0.09	0.5848	0.6493	0.5288	0.9067
	0.10	0.5263	0.6923	0.4932	0.9070
50%	0.05	0.9006	0.3918	0.7197	0.7243
	0.06	0.8538	0.5214	0.6540	0.8218
	0.07	0.7778	0.5859	0.6000	0.8485
	0.08	0.6608	0.5916	0.5539	0.9030
	0.09	0.5965	0.6456	0.5242	0.9082
	0.10	0.5380	0.6815	0.4877	0.9085

Tabela 4.4: Resultados obtidos pelo Algoritmo 4 para $m = 7$.

T6. [Incremento i .] Atribua $i \leftarrow i + 1$.

T7. [Fim do vídeo?] Se $i = n$, o algoritmo termina. Caso contrário, volte a T3. ■

As Tabelas 4.4 e 4.5 apresentam os resultados obtidos pelo Algoritmo 4, com $m = 7$ e $m = 3$ respectivamente, para vários valores de k_{grad} e os dois valores de k_{corte}^ψ com melhores resultados na detecção de cortes. Como advertido pelos autores do *twin-comparison* original [ZKS93], o valor de $z^1(i)$ pode cair abaixo de k_{grad} durante uma transição, o que levaria à detecção de duas tomadas no lugar de uma. Zhang *et al.* sugerem permitir algumas iterações de tolerância no passo T5, caso a diferença caia abaixo do limiar. Na solução aqui apresentada, optou-se por uma etapa de pós-processamento na qual dissoluções a menos de 8 quadros (aproximadamente um quarto de segundo) de distância uma da outra são fundidas em um única transição, o que eleva consideravelmente a precisão do Algoritmo 4.

Entretanto, o método não pode gerenciar corretamente todas as ocorrências de fatores extremos no vídeo. A Figura 4.7 ilustra uma situação em que um grande objeto se move na tomada, ocluindo grandes áreas dos quadros. *Zooms* são também outra fonte comum de falsos positivos.

k_{corte}	k_{grad}	$\mathcal{R}$	$\mathcal{P}$	$\mathcal{R}_{\text{cov}}$	$\mathcal{P}_{\text{cov}}$
40%	0.05	0.8830	0.3728	0.7225	0.7176
	0.06	0.8538	0.4983	0.6562	0.8357
	0.07	0.7778	0.5757	0.5975	0.8569
	0.08	0.6608	0.5825	0.5488	0.8971
	0.09	0.5848	0.6493	0.5252	0.9105
	0.10	0.5204	0.6794	0.4924	0.9103
50%	0.05	0.8889	0.3781	0.7255	0.6933
	0.06	0.8538	0.5087	0.6745	0.8187
	0.07	0.7778	0.5859	0.6073	0.8438
	0.08	0.6667	0.5876	0.5590	0.8958
	0.09	0.6023	0.6560	0.5235	0.9088
	0.10	0.5380	0.6866	0.4899	0.9089

Tabela 4.5: Resultados obtidos pelo Algoritmo 4 para $m = 3$.

Figura 4.7: Falso-positivo na detecção de transições graduais gerado por movimentação de um objeto de grandes dimensões em relação ao quadro.

Capítulo 5

Resultados experimentais

Os vídeos utilizados neste trabalho são parte do material fornecido pelo NIST aos participantes do TRECVID'04, na modalidade de delimitação de tomadas. Tais vídeos foram obtidos pelo Linguistic Data Consortium (LDC) no primeiro semestre do ano de 1998, a partir dos canais norte-americanos CNN e ABC, e armazenados em MPEG-1. Mais exatamente, cada vídeo corresponde a uma seqüência de aproximadamente 30 minutos dos noticiários *CNN Headline News* e *ABC World News Tonight*, incluindo os comerciais entre os blocos de programa. O material utilizado aqui soma cerca de 3 horas e 50 minutos de vídeo. Seu conjunto de transições é exibido na Tabela 5.1.

5.1 Análise de parâmetros

Três vídeos, CNN 03/02/1998, CNN 22/02/1998 e CNN 15/05/1998 foram utilizados na comparação do desempenho obtido pelo Algoritmo 4 para várias configurações diferentes de parâmetros: os limiares k_{corte}^{ψ} , k_{grad} , a janela de filtragem $2m + 1$ e duas medidas de variação entre quadros sucessivos. As medidas utilizadas foram a interseção de histogramas $z_{\cap}^1(i)$, já vista anteriormente, e a *correlação de histogramas*, definida por

$$z_{\circ}^1(i) = 1 - \frac{1}{3L} \sum_{c \in \{R, G, B\}} \sum_{l=0}^{L-1} h_i^c[l] \cdot h_j^c[l]. \quad (5.1)$$

Vídeo	Cortes	Dissoluções	<i>Fades</i>	Outras	Duração (min.)	Quadros
CNN 03/02/1998	280	124	14	33	31	56716
CNN 22/02/1998	309	69	11	21	29	52735
CNN 15/05/1998	283	70	17	44	28	50254
CNN 31/05/1998	359	71	10	27	28	50275
ABC 24/02/1998	296	91	14	26	28	51205
ABC 12/04/1998	345	93	16	28	29	51877
ABC 25/04/1998	295	141	18	21	28	51814
ABC 19/06/1998	321	90	16	21	28	51244
Totais	2488	749	116	221	229	416120

Tabela 5.1: Conjunto de testes, fornecido pelo LDC/NIST para o TRECVID'04, utilizado neste trabalho. A coluna *Fades* se refere a *fades-out* seguidos imediatamente por *fades-in*.

As seções a seguir apresentam os resultados obtidos na variação de cada parâmetro. Os valores que apresentaram o melhor desempenho foram utilizados na delimitação de tomadas dos outros cinco vídeos.

5.1.1 Base para a distância relativa

Quatro valores diferentes foram utilizados em testes para m : 3, 5, 7 e 11. A Tabela 5.2 exhibe os resultados obtidos utilizando-se $z_{\cap}^1(i)$, enquanto que a Tabela 5.3 mostra o desempenho aplicando-se $z_{\circ}^1(i)$.

Pode-se observar que o maior suporte temporal, dado por valores maiores de m , incrementa a precisão, reduzindo pela distância relativa a influência das movimentações intra-tomadas. Os valores $m = 7$ e $m = 11$ apresentam os melhores resultados. Nota-se contudo, em menor grau, que há supressão de cortes reais com o aumento do suporte temporal.

5.1.2 Histogramas e número de *bins*

Como visto anteriormente na Seção 3.1.4, os histogramas utilizados no cômputo da diferença entre quadros podem ter um número de *bins* menor do que L . Zhang *et al.* [ZKS93], por exemplo, utilizam histogramas com 64 *bins*. Esta estratégia diminui o tempo de processamento da medida, mas quais são seus efeitos no desempenho dos algoritmos?

Vídeo	m	$\mathcal{R}_{\text{corte}}$	$\mathcal{P}_{\text{corte}}$
CNN 03/02 1998	3	0.7928	0.6727
	5	0.7536	0.8508
	7	0.7464	0.8745
	11	0.7214	0.8978
CNN 22/02/1998	3	0.9352	0.5873
	5	0.8932	0.8817
	7	0.8673	0.9146
	11	0.8511	0.9228

Tabela 5.2: Resultados obtidos pelo Algoritmo 4 com $k_{\text{corte}} = 50\%$ para vários valores de m .

Vídeo	m	$\mathcal{R}_{\text{corte}}$	$\mathcal{P}_{\text{corte}}$
CNN 03/02 1998	3	0.8143	0.2744
	5	0.7786	0.4274
	7	0.7678	0.5071
	11	0.7536	0.5627
CNN 22/02/1998	3	0.9158	0.1983
	5	0.8835	0.3541
	7	0.8511	0.3931
	11	0.8317	0.4262

Tabela 5.3: Resultados obtidos pelo Algoritmo 4 com $k_{\text{corte}} = 50\%$ para vários valores de m .

Vídeo	$bins$	$\mathcal{R}_{\text{corte}}$	$\mathcal{P}_{\text{corte}}$	$\mathcal{R}_{\text{grad}}$	$\mathcal{P}_{\text{grad}}$
CNN 03/02 1998	64	0.7357	0.8803	0.8070	0.6667
	256	0.7214	0.8978	0.7953	0.5812
CNN 22/02/1998	64	0.8511	0.9100	0.6436	0.4710
	256	0.8511	0.9228	0.7129	0.4965
CNN 15/05/1998	64	0.8516	0.8486	0.7099	0.5886
	256	0.8198	0.7891	0.6870	0.5555

Tabela 5.4: Resultados obtidos pelo Algoritmo 4 sobre $z_{\cap}^1(i)$ com $k_{\text{corte}}^{\psi} = 50\%$, $k_{\text{grad}} = 0.04$ e $m = 11$. Foram utilizados histogramas com 64 e 256 $bins$.

Vídeo	$bins$	$\mathcal{R}_{\text{corte}}$	$\mathcal{P}_{\text{corte}}$	$\mathcal{R}_{\text{grad}}$	$\mathcal{P}_{\text{grad}}$
CNN 03/02 1998	64	0.7571	0.4988	0.5146	0.7586
	256	0.7536	0.5627	0.5263	0.7438
CNN 22/02/1998	64	0.8511	0.4361	0.3861	0.5735
	256	0.8317	0.4262	0.3960	0.5555
CNN 15/05/1998	64	0.8551	0.5021	0.4733	0.7294
	256	0.8339	0.4729	0.4580	0.7059

Tabela 5.5: Resultados obtidos pelo Algoritmo 4 sobre $z_{\circ}^1(i)$ com $k_{\text{corte}}^{\psi} = 50\%$, $k_{\text{grad}} = 0.04$ e $m = 11$. Foram utilizados histogramas com 64 e 256 $bins$.

Segundo os resultados apresentados na Tabela 5.4, a redução no número de $bins$ aparentemente não tem impacto no desempenho do Algoritmo 4 sobre $z_{\cap}^1(i)$. O reconhecimento e a precisão na detecção de cortes e transições graduais mostram-se muito próximas nos dois casos. O mesmo se constata com relação à correlação $z_{\circ}^1(i)$, como pode ser visto na Tabela 5.5.

5.1.3 Separação de tomadas: o limiar k_{corte}^{ψ}

O parâmetro k_{corte}^{ψ} , como visto anteriormente no Capítulo 4, estabelece um limiar para a distância relativa entre o máximo local e os demais valores na janela. Quanto maior o valor de k_{corte}^{ψ} , mais destacado deve ser $z^1(i)$ para que q_i delimite uma tomada.

As Tabelas 5.6 e 5.7 mostram como o valor de k_{corte}^{ψ} afeta os resultados obtidos pelo Algoritmo 4 sobre $z_{\cap}^1(i)$ e $z_{\circ}^1(i)$ respectivamente. Estes resultados mostram que valores

Vídeo	k_{corte}^{ψ}	$\mathcal{R}_{\text{corte}}$	$\mathcal{P}_{\text{corte}}$	$\mathcal{R}_{\text{grad}}$	$\mathcal{P}_{\text{grad}}$
CNN 03/02 1998	30	0.7893	0.6190	0.8304	0.6228
	40	0.7678	0.7544	0.8304	0.6604
	50	0.7357	0.8803	0.8070	0.6667
	60	0.6821	0.9409	0.8011	0.7061
	70	0.5750	0.9415	0.7602	0.7143
	80	0.4321	0.9453	0.6608	0.7290
CNN 22/02/1998	30	0.8867	0.5362	0.6337	0.3636
	40	0.8770	0.7324	0.6535	0.4049
	50	0.8511	0.9100	0.6436	0.4710
	60	0.8285	0.9624	0.6238	0.5122
	70	0.7508	0.9667	0.5742	0.5631
	80	0.5534	0.9827	0.4950	0.6493
CNN 15/05/1998	30	0.8869	0.6354	0.7328	0.5189
	40	0.8728	0.7553	0.7328	0.5454
	50	0.8516	0.8486	0.7099	0.5886
	60	0.7880	0.9330	0.6870	0.6475
	70	0.6714	0.9743	0.6107	0.6452
	80	0.4700	0.9708	0.4656	0.6630

Tabela 5.6: Resultados obtidos pelo Algoritmo 4 sobre $z_{\cap}^1(i)$ com $k_{\text{grad}} = 0.04$, $m = 11$ e 64 *bins* para vários valores de k_{corte}^{ψ} .

acima de 50% degeneram o reconhecimento, sobretudo o de cortes.

5.1.4 Transições graduais e o limiar k_{grad}

O Algoritmo 2 difere pouco do *twin-comparison* original na detecção de transições graduais. Um limiar k_{grad} busca identificar possíveis seqüências de quadros candidatos e, por fim, a diferença acumulada entre os quadros inicial e final de cada seqüência é analisada de forma similar à detecção de cortes.

Valores baixos de k_{grad} , além de gerarem falsos positivos, aumentam o tempo total de processamento devido a maior ocorrência de candidatos. Inversamente, valores altos podem perder transições, mas reduzem o tempo de execução do algoritmo.

As Tabelas 5.8 e 5.9 apresentam os resultados obtidos por vários valores de k_{grad} (com k_{corte}^{ψ} fixo em 50%). Movimentação de câmeras e objetos, *zooms* e transições graduais entre toma-

Vídeo	k_{corte}^{ψ}	$\mathcal{R}_{\text{corte}}$	$\mathcal{P}_{\text{corte}}$	$\mathcal{R}_{\text{grad}}$	$\mathcal{P}_{\text{grad}}$
CNN 03/02 1998	30	0.7786	0.2781	0.4795	0.7130
	40	0.7643	0.3919	0.4971	0.7203
	50	0.7571	0.4988	0.5146	0.7586
	60	0.7321	0.6137	0.5205	0.7672
	70	0.7000	0.7567	0.5029	0.7478
	80	0.6536	0.8512	0.4737	0.8020
CNN 22/02/1998	30	0.8835	0.2103	0.3861	0.5821
	40	0.8641	0.3034	0.3960	0.5714
	50	0.8511	0.4361	0.3861	0.5735
	60	0.8284	0.5447	0.3762	0.5672
	70	0.8123	0.6711	0.3762	0.5846
	80	0.7767	0.8191	0.3564	0.6207
CNN 15/05/1998	30	0.8939	0.2694	0.4580	0.8108
	40	0.8728	0.3847	0.4809	0.7590
	50	0.8551	0.5020	0.4733	0.7294
	60	0.8162	0.6193	0.4962	0.7222
	70	0.7668	0.7457	0.4809	0.7159
	80	0.6996	0.8684	0.4504	0.7284

Tabela 5.7: Resultados obtidos pelo Algoritmo 4 sobre $z_{\circ}^1(i)$ com $k_{\text{grad}} = 0.04$, $m = 11$ e 64 *bins* para vários valores de k_{corte}^{ψ} .

Vídeo	k_{corte}^{ψ}	$\mathcal{R}_{\text{grad}}$	$\mathcal{P}_{\text{grad}}$
CNN 03/02 1998	0.04	0.8070	0.6667
	0.05	0.7544	0.6898
	0.06	0.6959	0.7484
	0.07	0.6257	0.7589
CNN 22/02/1998	0.04	0.6435	0.4710
	0.05	0.5742	0.5088
	0.06	0.5148	0.5252
	0.07	0.4851	0.5764
CNN 15/05/1998	0.04	0.7099	0.5886
	0.05	0.6564	0.6056
	0.06	0.6183	0.6807
	0.07	0.5725	0.7211

Tabela 5.8: Resultados obtidos pelo Algoritmo 4 sobre $z_{\cap}^1(i)$ com $k_{\text{corte}}^{\psi} = 50\%$, $m = 11$ e 64 *bins* para vários valores de k_{grad} .

das com distribuição similar de luminosidade são as principais causas do baixo desempenho apresentado.

5.2 Testes

Os oito vídeos foram processados pelo Algoritmo 4 sobre $z_{\cap}^1(i)$, calculado em histogramas de 64 *bins*, utilizando-se os três melhores valores de k_{corte}^{ψ} e k_{grad} . O suporte temporal foi dado por $m = 11$.

Os resultados são exibidos nas Tabelas 5.10 e 5.11. O desempenho alcançado na detecção de cortes é próximo de 90% tanto em reconhecimento quanto em precisão. Já o desempenho obtido na detecção de transições graduais é inferior, girando em torno de 70% para $\mathcal{R}_{\text{grad}}$ e $\mathcal{P}_{\text{grad}}$. Contudo, este resultado é comumente aceitável em aplicações de sumarização de vídeo, como a que veremos a seguir.

Vídeo	k_{corte}^{ψ}	$\mathcal{R}_{\text{grad}}$	$\mathcal{P}_{\text{grad}}$
CNN 03/02 1998	0.02	0.6667	0.6909
	0.03	0.5672	0.7239
	0.04	0.5146	0.7586
	0.05	0.4561	0.7573
	0.06	0.4152	0.7717
	0.07	0.3567	0.7721
CNN 22/02/1998	0.02	0.5544	0.4786
	0.03	0.4257	0.5308
	0.04	0.3861	0.5735
	0.05	0.3960	0.625
	0.06	0.3861	0.7091
	0.07	0.3564	0.6923
CNN 15/05/1998	0.02	0.6183	0.6000
	0.03	0.5420	0.6961
	0.04	0.4733	0.7294
	0.05	0.4275	0.7778
	0.06	0.3893	0.7612
	0.07	0.3511	0.7419

Tabela 5.9: Resultados obtidos pelo Algoritmo 4 sobre $z_o^1(i)$ com $k_{\text{corte}}^{\psi} = 50\%$, $m = 11$ e 64 *bins* para vários valores de k_{grad} .

Vídeo	k_{corte}^{ψ}	$\mathcal{R}_{\text{corte}}$	$\mathcal{P}_{\text{corte}}$	k_{grad}	$\mathcal{R}_{\text{grad}}$	$\mathcal{P}_{\text{grad}}$	$\mathcal{R}_{\text{cov}}$	$\mathcal{P}_{\text{cov}}$
CNN 03/02/1998	40%	0.7678	0.7544	0.04	0.8304	0.6605	0.7052	0.8446
				0.05	0.7719	0.6984	0.7167	0.9029
				0.06	0.7017	0.7453	0.6072	0.9270
	50%	0.7357	0.8803	0.04	0.8070	0.6667	0.7119	0.8612
				0.05	0.7544	0.6898	0.7218	0.9037
				0.06	0.6959	0.7484	0.6468	0.9274
	60%	0.6821	0.9409	0.04	0.8012	0.7062	0.7638	0.8727
				0.05	0.7602	0.7558	0.7051	0.9190
				0.06	0.6667	0.7917	0.6470	0.9265
CNN 22/02/1998	40%	0.8770	0.7324	0.04	0.6535	0.4049	0.6372	0.8789
				0.05	0.5940	0.4800	0.5960	0.9046
				0.06	0.1287	0.5909	0.6990	0.9096
	50%	0.8511	0.9100	0.04	0.6436	0.4710	0.6677	0.8445
				0.05	0.5742	0.5088	0.6073	0.9041
				0.06	0.5148	0.5252	0.6000	0.9425
	60%	0.8285	0.9624	0.04	0.6238	0.5122	0.6515	0.8409
				0.05	0.5544	0.5544	0.6205	0.9165
				0.06	0.4752	0.6076	0.6004	0.9474
CNN 15/05/1998	40%	0.8728	0.7553	0.04	0.7328	0.5454	0.7166	0.8184
				0.05	0.6641	0.5918	0.6523	0.8415
				0.06	0.6412	0.6885	0.6274	0.9001
	50%	0.8516	0.8486	0.04	0.7099	0.5886	0.7260	0.8206
				0.05	0.6565	0.6056	0.6547	0.8562
				0.06	0.6183	0.6807	0.6219	0.8923
	60%	0.7880	0.9330	0.04	0.6870	0.6475	0.7248	0.8163
				0.05	0.6259	0.6406	0.6469	0.8670
CNN 31/05/1998	40%	0.8914	0.7619	0.04	0.5833	0.5431	0.6667	0.7705
				0.05	0.5000	0.5400	0.6448	0.8821
				0.06	0.4722	0.5795	0.5701	0.9099
	50%	0.8774	0.8974	0.04	0.6111	0.6055	0.6664	0.7865
				0.05	0.5185	0.5714	0.6499	0.8920
				0.06	0.4815	0.6046	0.5760	0.9182
	60%	0.8468	0.9412	0.04	0.6204	0.6569	0.6861	0.7825
				0.05	0.5092	0.6180	0.6584	0.8830
				0.06	0.4722	0.6456	0.6088	0.9118

Tabela 5.10: Resultados obtidos pelo Algoritmo 4 sobre $z_{\cap}^1(i)$ com $m = 11$ no conjunto de testes oriundo do canal de televisão CNN.

Vídeo	k_{corte}^{ψ}	$\mathcal{R}_{\text{corte}}$	$\mathcal{P}_{\text{corte}}$	k_{grad}	$\mathcal{R}_{\text{grad}}$	$\mathcal{P}_{\text{grad}}$	$\mathcal{R}_{\text{cov}}$	$\mathcal{P}_{\text{cov}}$
ABC 24/02/1998	40%	0.8986	0.9268	0.04	0.7404	0.6831	0.6322	0.8922
				0.05	0.6488	0.7944	0.5744	0.9174
				0.06	0.5725	0.8152	0.5497	0.9516
	50%	0.8649	0.9481	0.04	0.7328	0.7111	0.6167	0.8942
				0.05	0.6564	0.7963	0.5651	0.9201
				0.06	0.5649	0.8222	0.5314	0.9576
	60%	0.8446	0.9506	0.04	0.6870	0.7692	0.6346	0.9170
				0.05	0.5878	0.8555	0.5854	0.9150
				0.06	0.5267	0.8625	0.5374	0.9560
ABC 12/04/1998	40%	0.9043	0.8455	0.04	0.7956	0.6687	0.5820	0.9008
				0.05	0.7080	0.7185	0.5765	0.9346
				0.06	0.6569	0.8036	0.5760	0.9634
	50%	0.8898	0.9136	0.04	0.7810	0.7230	0.6160	0.9079
				0.05	0.7008	0.7680	0.5996	0.9406
				0.06	0.6496	0.8318	0.6067	0.9631
	60%	0.8783	0.9619	0.04	0.7591	0.8000	0.6436	0.9093
				0.05	0.7153	0.8596	0.6147	0.9402
				0.06	0.6350	0.8877	0.6026	0.9613
ABC 25/04/1998	40%	0.8712	0.7859	0.04	0.8278	0.6367	0.7214	0.8110
				0.05	0.7611	0.7287	0.6850	0.8976
				0.06	0.7111	0.8050	0.6885	0.9511
	50%	0.8576	0.9200	0.04	0.8222	0.7014	0.7729	0.8128
				0.05	0.7444	0.7836	0.7687	0.8938
				0.06	0.7111	0.8366	0.6872	0.9443
	60%	0.8135	0.9562	0.04	0.7944	0.7566	0.7685	0.8073
				0.05	0.7055	0.8089	0.7519	0.8897
				0.06	0.6667	0.8276	0.6824	0.9460
ABC 19/06/1998	40%	0.9439	0.9352	0.04	0.8189	0.6265	0.7162	0.8635
				0.05	0.7401	0.7460	0.6663	0.9001
				0.06	0.6535	0.7477	0.6667	0.9384
	50%	0.9283	0.9613	0.04	0.8110	0.6867	0.7374	0.8875
				0.05	0.7244	0.7541	0.6991	0.9182
				0.06	0.6535	0.7615	0.6656	0.9582
	60%	0.8941	0.9762	0.04	0.7795	0.7279	0.7719	0.8880
				0.05	0.7008	0.7807	0.7289	0.9144
				0.06	0.6299	0.8000	0.6979	0.9646

Tabela 5.11: Resultados obtidos pelo Algoritmo 4 sobre $z_{\cap}^1(i)$ com $m = 11$ no conjunto de testes oriundo do canal de televisão ABC.

5.3 Aplicação: sumarização de vídeo digital

A Figura 5.1 mostra uma aplicação simples para a delimitação de tomadas: sumarização de vídeo para a exploração rápida de seu conteúdo por parte do usuário. O sumário, construído automaticamente com base nos resultados obtidos pelo Algoritmo 4, pode ser visto em um navegador *Web* comum.

Um quadro-chave é apresentado por tomada. A legenda abaixo do quadro informa o número da tomada e seu instante de início (minutos e segundos) no vídeo. A medida $z_{\cap}(q_i, q_j)$ é aplicada entre os quadros-chaves e comparada a um limiar, criando-se assim referências entre tomadas similares. A Figura 5.1 ilustra uma dessas ligações: a tomada 4 referencia a tomada 44 (as duas exibem o âncora do *ABC World News Tonight*).

Capítulo 6

Conclusões

Neste trabalho, foi apresentado o problema da segmentação temporal de tomadas e sua importância na estruturação e indexação de vídeo digital. As principais propostas de solução encontradas na literatura foram introduzidas e debatidas. Este estudo se concentrou em medidas de variação do conteúdo visual entre quadros do vídeo, baseadas em histogramas de cor.

A principal contribuição, nesse sentido, foi uma revisão do clássico algoritmo *twin-comparison* (Algoritmo 2) de Zhang *et al.* [ZKS93], propondo uma nova versão aqui chamada de *twin-comparison ψ* (Algoritmo 4). Tal versão se baseia em máximos locais, encontrados no sinal de variação do conteúdo visual, e sua distância relativa aos demais valores de sua vizinhança. Foi discutido, através de testes com várias seqüências de vídeo, como os parâmetros utilizados pelo algoritmo (limiares, suporte temporal, granularidade do histograma) impactam seu desempenho. Examinou-se, ainda, duas formas diferentes de comparação de histogramas: intersecção entre *bins* e correlação de histogramas, sendo a primeira medida a que apresentou melhores resultados, como defendido por Gargi *et al.* [GKS00].

Os resultados apresentados aqui e os encontrados na literatura recente mostram que, embora a detecção de cortes possa obter níveis razoáveis de reconhecimento e precisão para sua aplicação em sistemas de sumarização de vídeo, a detecção de transições graduais ainda não atinge níveis similares de desempenho. Características de *baixo-nível*, isto é, com pouca in-

formação semântica, tais como distribuição de cor, média e variância de luminosidade, podem assumir comportamentos similares ao longo de transições entre tomadas, movimentação de objetos, operações de câmera ou variação de luz, entre outras, o que dificulta a caracterização correta de cada um desses eventos.

A combinação dessas várias características na construção de um delimitador de tomadas pode levar a melhores resultados. Combinar características oriundas de domínios diferentes, como histogramas (domínio de cor) e variação entre pixels (domínio espacial) pode ser uma forma de compensar suas deficiências individuais.

Trabalhos recentes como os de Vasconcelos e Lippman [VL00] e Hanjalic [Han02] apontam na direção de classificadores bayesianos que utilizam não só as distribuições da variação do conteúdo visual em tomadas e entre tomadas como também probabilidades computadas com base na duração das tomadas. Tais classificadores podem se mostrar alternativas mais robustas aos inúmeros eventos presentes no decorrer do vídeo e que são fontes de falsos positivos e falsos negativos.

O estudo de soluções bayesianas (e outras abordagens probabilísticas), a composição de características de baixo-nível em descritores com maior poder de classificação e técnicas baseadas em aglomerados são os próximos passos para a continuação deste estudo, bem como a aplicação da delimitação de tomadas em novas ferramentas para navegação de exploração de vídeo digital.

Referências Bibliográficas

- [Ala93] Adnan M. Alattar. Detecting and compressing dissolve regions in video sequences with a dvi multimedia image compression algorithm. *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, 1:13–16, Maio 1993.
- [BB82] D. .H Ballard and C. M. Brown. *Computer Vision*. Prentice-Hall, New Jersey, NY, 1982.
- [BB95] S. S. Beauchemin and J. L. Barron. The computation of optical flow. *ACM Computing Surveys*, 27(3), Setembro 1995.
- [BDBP02] Marco Bertini, Alberto Del Bimbo, and Pietro Pala. Indexing for reuse of tv news shots. *Pattern Recognition*, 35:581–591, Março 2002.
- [BDF⁺99] P. Bouthemy, Y. Dufournaud, R. Fablet, R. Mohr, S. Peleg, and A. Zomet. Content-based browsing and navigation. In *European Workshop on Content-Based Multimedia Indexing CBMI'99*, Toulouse, France, Outubro 1999.
- [BR96] J. S. Boreczky and L. Rowe. Comparison of video shot boundary detection techniques. In *Proc. IS&T/SPIE Storage and Retrieval for Still Image and Video Databases IV*, volume 2670, pages 170–179, Fevereiro 1996.
- [Can86] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

- [CHH⁺00] Shih-Fu Chang, Qian Huang, Thomas Huang, Atul Puri, and Behzad Shahraray. Multimedia search and retrieval. In Atul Puri and Tsuhan Chen, editors, *Advances in Multimedia: Systems, Standards and Networks*. Marcel Dekker, New York, 2000.
- [DSP91] Glorianna Davenport, Thomas Aguierre Smith, and Natalio Pincever. Cinematic primitives for multimedia. *IEEE Computer Graphics and Applications*, 11:67–74, Julho 1991.
- [FT98] A. Müfit Ferman and A. Murat Tekalp. Efficient filtering and clustering methods for temporal video segmentation and visual summarization. *Journal of Visual Communication and Image Representation*, 9(4):336–351, Dezembro 1998.
- [GCAL03] Silvio Jamil Ferzoli Guimarães, Michel Couprie, Arnaldo de Albuquerque Araújo, and Neucimar Jerônimo Leite. Video segmantation based on 2d image analysis. *Pattern Recognition Letters*, 24:947–957, 2003.
- [GKS00] Ullas Gargi, Rangachar Kasturi, and Susan H. Strayer. Performance characterization of video-shot-change detection methods. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(1):1–13, 2000.
- [GW00] Rafael C. Gonzalez and Richard E. Woods. *Processamento de Imagens Digitais*. Edgard Blücher, 1 edition, 2000.
- [Han02] Alan Hanjalic. Shot-boundary detection: Unraveled and resolved? *IEEE Transactions on Circuits and Systems for Video Technology*, 12(2):90–105, fevereiro 2002.
- [HWJ94] A Hampapur, T. Weymouth, and R. Jain. Digital video segmentation. In *Proceedings of the second ACM international conference on Multimedia*, pages 357–364. ACM Press, 1994.

- [IP97] F Idris and S. Panchanathan. Review of image and video indexing techniques. *Journal of Visual Communication and Image Representation*, 8(2):146–166, Junho 1997.
- [IR98] ITU-R. Recommendation bt.470-6. Technical report, International Telecommunication Union, Novembro 1998.
- [KC01] Irena Koprinska and Sergio Carrato. Temporal video segmentation: A survey. *Signal Processing: Image Communication*, 16(5):477–500, Junho 2001.
- [KJ91] R. Kasturi and R. Jain. *Computer Vision: Principles*, chapter Dynamic Vision, pages 469–480. IEEE Computer Society Press, Washington, DC, 1991.
- [Lie99] Rainer Lienhart. Comparison of automatic shot boundary detection algorithms. In *Proc. IS&T/SPIE Storage and Retrieval for Image and Video Databases VII*, volume 3656, pages 290–301, 1999.
- [Lie01] Rainer Lienhart. Reliable transition detection in videos: A survey and practitioner’s guide. *International Journal of Image and Graphics*, 1(3):469–486, 2001.
- [LPE97] Rainer Lienhart, Silvia Pfeiffer, and Wolfgang Effelsberg. Video abstracting. *Communications of ACM*, 40(12):54–62, 1997.
- [MPEG02] MPEG Moving Picture Experts Group. Mpeg-4 overview - (v.21 - jeju version). Technical report, International Organisation for Standardisation, Março 2002.
- [MPEG03] MPEG Moving Picture Experts Group. Mpeg-7 overview (version 9). Technical report, International Organisation for Standardisation, Março 2003.
- [NT92] Akio Nagasaka and Yuzuru Tanaka. Automatic video indexing and full-video search for object appearances (abstract). *J. Inf. Process.*, 15(2):316, 1992.
- [Pro] The Open Video Project. <http://www.open-video.org>.

- [Sha95] B. Shahraray. Scene change detection and content-based sampling of video sequences. In *Proceedings of IS&T/SPIE*, volume 2419, pages 2–13, 1995.
- [SLS03] Nicu Sebe, Michael S. Lew, and Arnold W. M. Smeulders. Video retrieval and summarization: editorial introduction. *Computer Vision and Image Understanding*, 92(2-3):141–146, Novembro-Dezembro 2003.
- [TDV00] Ba Tu Truong, Chitra Dorai, and Svetha Venkatesh. New enhancements to cut, fade, and dissolve detection processes in video segmentation. In *Proceedings of the eighth ACM international conference on Multimedia*, pages 219–227. ACM Press, 2000.
- [VL97] Nuno Vasconcelos and Andrew Lippman. A bayesian video modeling framework for shot segmentation and content characterization. In *Proceedings of the 1997 Workshop on Content-Based Access of Image and Video Libraries (CBAIVL '97)*, page 59. IEEE Computer Society, 1997.
- [VL00] Nuno Vasconcelos and Andrew Lippman. Statistical models of video structure for content analysis and characterization. *IEEE Transactions on Image Processing*, 9(1):3–19, Janeiro 2000.
- [XL98] Wei Xiong and John Chung-Mong Lee. Efficient scene change detection and camera motion annotation for video classification. *Computer Vision and Image Understanding*, 71(2):166–181, Agosto 1998.
- [XLI95] Wei Xiong, John Chung-Mong Lee, and Man-Ching Ip. Net comparison: a fast and effective method for classifying image sequences. In *Proc. of SPIE Conference on Storage and Retrieval for Image and Video Databases III*, volume 2420, pages 318–328, San Jose, CA, 1995.
- [YL95] Boon-Lock Yeo and Bede Liu. Rapid scene analysis on compressed video. *IEEE Transactions on Circuits and Systems for Video Technology*, 5:533–544, Dezembro 1995.

- [ZCPR03] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, 2003.
- [ZKS93] HongJiang Zhang, Atreyi Kankanhalli, and Stephen W. Smoliar. Automatic partitioning of full-motion video. *Multimedia Systems*, 1(1):10–28, 1993.
- [ZMM95] Ramin Zabih, Justin Miller, and Kevin Mai. A feature-based algorithm for detecting and classifying scene breaks. *Proceedings of ACM Multimedia 95*, pages 189–200, Novembro 1995.
- [ZSW95] Chien Young Zhang, Hong Jiang Low, Stephen W. Smoliar, and Jian Hua Wu. Video parsing, retrieval and browsing: An integrated and content-based solution. In *ACM Multimedia 95*, San Francisco, California, Novembro 1995.