

CENTRO UNIVERSITÁRIO DA FEI

MAURÍCIO KENITI HIROTA
FABRÍCIO LOPES DE SOUZA
IVAN SHIGUENORI MACHIDA
THIAGO MIZUTANI

Orientador: Prof. D.Sc. Paulo Sérgio Silva Rodrigues

**M-FIT Motion Finder Integration :
Editoração Eletrônica de Vídeos**

São Bernardo do Campo
2008

CENTRO UNIVERSITÁRIO DA FEI

MAURÍCIO KENITI HIROTA
FABRÍCIO LOPES DE SOUZA
IVAN SHIGUENORI MACHIDA
THIAGO MIZUTANI

Orientador: Prof. D.Sc. Paulo Sérgio Silva Rodrigues

M-FIT Motion Finder Integration : Editoração Eletrônica de Vídeos

Trabalho de Conclusão de curso, apresentado no Centro Universitário da FEI, como parte dos requisitos necessários para obtenção do título de Bacharel em Ciências da Computação, orientado pelo professor Prof. D.Sc. Paulo Sérgio Silva Rodrigues.

São Bernardo do Campo

2007

Resumo

Na área de Editoração de Vídeos, uma das maiores demandas é a detecção de transição de tomadas, que é basicamente o corte de uma tomada e o início de outra, o que é útil para diversas aplicações, como indexação, recuperação, análise e edição de vídeos. Estas transições podem ocorrer de diversas formas, tais como: zooms, cortes, fades, dissolves. Este trabalho apresenta o M-FIT Motion Finder Integration, um framework que possibilitará o reconhecimento automático das transições de tomadas de cenas a partir de uma determinada amostragem de imagens (frames de um filme) através do uso de técnicas de Ritmo Visual e Morfologia Matemática, permitindo que o ponto exato da transição seja obtido de uma maneira mais prática e rápida. O M-FIT é uma ferramenta desenvolvida em linguagem C, com utilização de OpenCV (biblioteca para computação visual). Sua interface é feita com a utilização da ferramenta QT, tornando o M-FIT um sistema com uma interface simples e de fácil entendimento, a qual terá diversos atalhos para outras funcionalidades do sistema.

Conteúdo

1	Introdução	1
1.1	Objetivo	2
2	Trabalhos Relacionados	3
2.1	Detecção de Transições	3
2.2	Técnicas de PDI	5
3	Revisão Bibliográfica	7
3.1	Edição de Vídeo	8
3.1.1	Fade-in	8
3.1.2	Fade-out	8
3.1.3	Corte	9
3.1.4	Dissolve	9
3.2	Sistemas de Vídeo	10
3.3	Sistemas de Cor	12
3.3.1	RGB	12

Lista de Figuras

3.1	Exemplo de fade-in em uma sequencia de frames. Como pode ser observado não se sabe exatamente onde ocorre a transição.	8
3.2	Exemplo de Fade-out em uma seqüência de frames	8
3.3	Exemplo de corte em uma seqüência de frames.	9
3.4	Exemplo de Dissolve em seqüência de frames.	9
3.5	Sistema de Codificação e Decodificação de um vídeo no formato MPEG. . .	10
3.6	Sistema de armazenamento AVI.	10
3.7	Sistema RGB.	12

Capítulo 1

Introdução

O setor de entretenimento está sempre em constantes mudanças e inovações. Isto se deve ao fato do dele estar diretamente ligado a muitos outros setores, como o setor de marketing e cinema. Devido a isto, o número de novos filmes, propagandas, vêm crescendo rapidamente. Entre uma das tarefas mais demandadas da indústria da produção de vídeo, está a edição, que consiste na transformação de um vídeo através da aplicação de filtros e aplicação de efeitos como, por exemplo, ajustes de cor, brilho, contraste e remoção de objetos. Com isso a edição de vídeo torna-se uma atividade com um mercado amplo e que gera uma boa renda anualmente. Um vídeo normalmente é formado por um conjunto de sub-vídeos, que são conhecidos como tomadas. Em uma pré-edição, esses sub-vídeos são agrupados formando o vídeo completo. Todas estas tomadas são interligadas através de efeitos chamados transições, as quais podem ser de vários tipos, desde um simples corte, até uma transição mais complexa como fade ou dissolve. Estas transições são inseridas no vídeo com o objetivo de suavizar as trocas de tomadas. O problema de uma edição de vídeo é que existem várias tomadas em um único vídeo, então se o editor desejar ajustar o brilho de uma única tomada, por exemplo, terá que delimitar o seu início e fim manualmente. Sendo assim, a detecção e localização de transição é uma das tarefas mais importantes no processo de indexação e edição de vídeo digital. Este processo é trabalhoso e demorado, tornando necessária cada vez mais a produção de sistemas automáticos para detecção de transições. Neste trabalho será apresentado um sistema de detecção automática de transições de vídeo baseado em ritmo visual e morfologia matemática.

1.1 Objetivo

Desenvolver um sistema automático de detecção de transição de tomadas e edição de vídeo.

Capítulo 2

Trabalhos Relacionados

2.1 Detecção de Transições

Uma das técnicas mais comuns e importantes na área de editoração de vídeos é a detecção de transições.

Silvio J. F. Guimarães et al., 2003, propuseram um método para fazer a detecção de transições em um vídeo, aplicando o operador gradiente multi-escalar morfológico em cima de uma determinada amostragem do vídeo. Através deste método, puderam obter bons resultados através do gradiente multi-escalar em relação à detecção de transições graduais, porém o método não foca na dimensão da transição.

Silvio J. F. Guimarães et al., 2004, propuseram um método para detectar transições graduais (fade,dissolve) utilizando ritmo visual e morfologia matemática . O foco principal é usar operadores de morfologia matemática para aumentar os vestígios de transições na imagem do ritmo visual, não alterando os já explícitos. Com a aplicação deste método obtiveram uma taxa de acerto de 93.5

Silvio. J. F. Guimarães et al., 2002, com o objetivo de realizar a identificação e classificação de transições graduais entre duas zonas planas consecutivas (conjunto de pontos consecutivos com os mesmos tons de cinza), aplicaram um método baseado na decomposição hierárquica dos componentes da imagem em resíduos morfológicos. Com o uso deste método era possível retirar as informações necessárias sobre a transição e ainda fazer a reconstrução exata da imagem a partir dos seus resíduos. Desta forma, conseguiram fazer a detecção e identificação das transições facilmente. Apesar de o método funcionar bem em relação às transições graduais, o trabalho não faz referência às transições bruscas

(cortes).

Silvio Jamil Ferzoli Guimarães et al., 2001, desenvolveram um método para a detecção de cortes de vídeos digitais, a partir da aplicação de ritmo visual. Com este método, conseguiram obter com sucesso as transições de corte de um vídeo. A partir deste estudo, é possível desenvolver métodos para detecções de outros tipos de transições.

Nielsen Cassiano Simões, 2004, apresentou um método para detecção de transições abruptas (cortes e flashes) em vídeos, baseado em algoritmos de identificação de padrão em sinais 1D. O método proposto possibilitou a detecção de transições abruptas, ficando aberto a novas propostas para a detecção de outros tipos de transição.

C. W. Ngo et al., 1999, com a intenção de fazer a detecção de transições do tipo corte, wipe e dissolve, propuseram um método baseado em Ritmo Visual. Com a aplicação deste método, foi possível fazer a detecção e classificação das transições de uma maneira eficiente. Ressalta-se ainda que futuramente, pretende-se fazer uma análise de takes de vídeos para fazer indexação de vídeos e recuperação de dados.

Silvio J. F. Guimarães et al., 2002, propuseram um método que faz uso de Ritmo Visual (transformação dos eventos do vídeo em padrões de imagens 2D), juntamente com a aplicação de operadores de Morfologia Matemática e Topologia Digital para fazer a detecção e identificação dos eventos de transição de um vídeo. Aplicando este método, puderam fazer a detecção de fades curtos com uma baixa taxa de erros, sendo possível ainda desenvolver uma extensão do método para detecção de outros tipos de transições, assim como fazer a aplicação do método diretamente nos dados do vídeo.

Sébastien Stevie et al, 2003, apresentaram uma gama de algoritmos referentes a identificação de transições em vídeos sem compressão utilizando métodos baseados em: pixel, histograma, bloco, características, movimento, entre outros. O objetivo era analisar o custo computacional de cada algoritmo e compará-los. Concluiu-se que os algoritmos que utilizam lógicas simples geralmente possuem um custo computacional baixo, porém dispõem de uma qualidade menor nas detecções. Em contrapartida, algoritmos que utilizam lógicas mais complexas, como a análise de movimento e característica, ou um método que use vários algoritmos, conseguem ter como resultado final com boa qualidade em termos de detecção de transição. Porém, possuem um custo computacional mais alto.

Guimarães, S. J. F. et al, 2001, propuseram um método que transforma o problema de segmentação de vídeo em um problema de detecção de padrões , onde cada efeito de

vídeo é transformado em diferentes padrões de ritmo visual. Obtiveram bons resultados em comparação à outros métodos em relação a robustez, falhas e medidas de gama, porém foram piores que outros métodos em relação a quantidade de detecções quando não há nenhuma falsa-deteção (falseless error).

Silvio Jamil Ferzoli Guimarães et al, 2002, propuseram um método para detecção de transições parametrizadas e direcionais, baseado em resíduos morfológicos considerando elementos linearmente estruturados. Com o método proposto, puderam identificar facilmente as transições, fato que não é possível, por exemplo, com o método que utiliza o gradiente Soille [7].

Com o objetivo de identificar o limite entre tomadas consecutivas de um determinado segmento de vídeo, S. J. F. Guimarães et al., 2002, utilizaram um método baseado no ritmo visual e Morfologia matemática, onde cada evento de vídeo é representado por um padrão em uma imagem 2D. Através da aplicação destes métodos, puderam identificar facilmente as transições de um segmento de vídeo, porém foram somente aplicadas aos efeitos de transições fades, cortes e dissolves.

Com o objetivo de identificar cortes de uma cena utilizando um único algoritmo, Jianmin Jiang et al., 2005, propuseram a aproximação da lógica Fuzzy para integrar funcionalidades híbridas para a detecção de cortes de uma cena dentro de um vídeo aleatório. A aproximação da lógica Fuzzy contém dois modos de processamento, onde um é dedicado à detecção de cortes bruscos de cenas e também transições dissolves, e o outro para transições graduais. Estes dois modos são unificados por um seletor de modos que decide qual modo o esquema deve trabalhar para atingir o melhor desempenho de detecção, sendo que não se aplicam a transições flash.

2.2 Técnicas de PDI

Para realizar a detecção de transições ou qualquer tipo de editoração de vídeos, são necessárias técnicas de PDI como Morfologia Matemática, Entropia, Filtragem Espacial, Detecção de Bordas, entre outras.

Com o objetivo de segmentar imagens, Gilles Bertrand et al., 1999, utilizaram transformações de níveis e homotópicas (estudos dos pontos de cruzamento da topologia e reconstrução da transformação). Utilizando estas técnicas conseguiram segmentar ima-

gens em tons de cinza, sendo que estas não se aplicam a imagens RGB.

Com o objetivo de analisar a morfologia de imagens em 3D, K. Michielsen et al., 2000, utilizaram o corte dos lados de um cubo 3D e aplicaram a morfologia matemática e a tripla periódica mínima que verifica a continuidade de uma superfície. Com estas técnicas, conseguiram caracterizar a morfologia de uma imagem 3D em tons de cinza, sendo que não se aplicam a imagens em RGB.

Com o objetivo de detectar cortes de cenas e otimizar o custo de tempo humano, Jun Yu et al., 2000, utilizaram a diferença de métrica baseado no histograma de cores de sucessivos frames de uma sequência de vídeo utilizando a entropia de um threshold, onde o threshold é a base da diferença dos histogramas de tons de cinza de um determinado frame. Por outro lado, para confirmar os resultados, são necessárias interações humanas e um procedimento de eliminação de falsa detecção futuramente será desenvolvido para melhorar a precisão.

Capítulo 3

Revisão Bibliográfica

Nesta seção, serão definidos os principais conceitos para o entendimento de todo o projeto e a metodologia do sistema proposto.

3.1 Edição de Vídeo

A edição de vídeos consiste em qualquer operação que modifique o conteúdo original de um vídeo. Tais operações podem consistir desde a segmentação do vídeo para futuras modificações de indexação ou armazenamento, até a aplicação de efeitos visuais, como por exemplo, uma mudança de cores, o nivelamento da luminosidade entre frames de uma tomada, eliminação de ruídos, equalização de som com imagem, entre outros, os quais são intensamente utilizados no setor cinematográfico, de pesquisa e entretenimento. É comum organizar a edição de vídeos através de tomadas, as quais são divididas por transições. Portanto, é importante que seja feita a detecção destas transições antes da edição do vídeo para que um determinado efeito aplicado ou qualquer outra operação de edição possa ser delimitado às tomadas específicas e não para o vídeo todo.

3.1.1 Fade-in

O fade-in é uma transição relativamente lenta e suave entre o preto e uma imagem qualquer. Este tipo de transição geralmente é utilizado no início do vídeo ou no início de um bloco de tomadas.

Podemos visualizar o fade-in na figura 3.1:

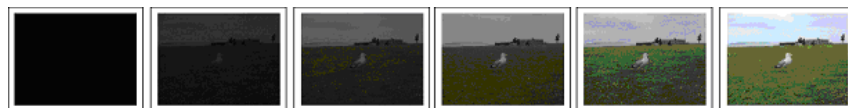


Figura 3.1: Exemplo de fade-in em uma sequência de frames. Como pode ser observado não se sabe exatamente onde ocorre a transição.

3.1.2 Fade-out

Assim como o fade-in, o fade-out também é uma transição relativamente lenta e suave, porém com um efeito oposto. Um exemplo desta transição pode ser observado na Figura 3.2:

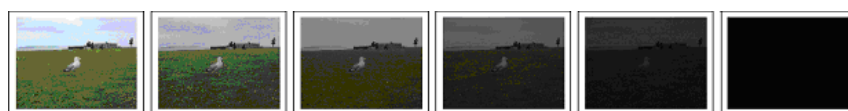


Figura 3.2: Exemplo de Fade-out em uma sequência de frames

3.1.3 Corte

Cortes são transições bruscas e secas entre tomadas, onde o último frame de uma determinada tomada é concatenado ao primeiro frame da tomada seguinte diretamente. A Figura 3.3 mostra a ocorrência de um corte:

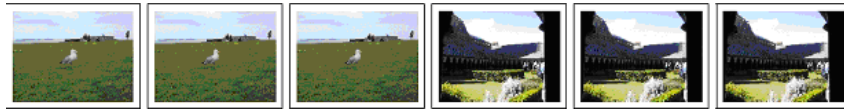


Figura 3.3: Exemplo de corte em uma sequência de frames.

Note que no exemplo da Figura 3.3 é possível se ver claramente que a transição ocorre entre o terceiro e quarto frame.

3.1.4 Dissolve

Dissolves O dissolve é uma transição que mistura fade-out e fade-in, isto é, a primeira tomada dá lugar a uma segunda tomada gradualmente, havendo uma sobreposição dos primeiros frames da próxima tomada sobre os últimos da tomada anterior. Geralmente, este tipo de transição é usado na mudança de paisagens. Podemos visualizar este efeito na Figura 3.4:



Figura 3.4: Exemplo de Dissolve em sequência de frames.

Pode-se ver na Figura 3.4 a primeira tomada dando lugar gradativamente para a segunda tomada até que não haja mais sobreposição de imagens. Uma boa abordagem sobre a definição do dissolve pode ser encontrada em [28].

3.2 Sistemas de Vídeo

Os sistemas de vídeo consistem na organização em que uma determinada informação de vídeo é armazenada em forma digital, tais como AVI, MPEG, entre outros. O MPEG (Movie Pictures Experts Group) é um padrão internacional definido pela ISO que tem como característica a compressão de um vídeo. Onde ao comprimir uma determinada amostragem de um vídeo, o mesmo é passado pelo canal compressor, e ao exibir esta mesma amostragem o vídeo passa pelo canal expensor.

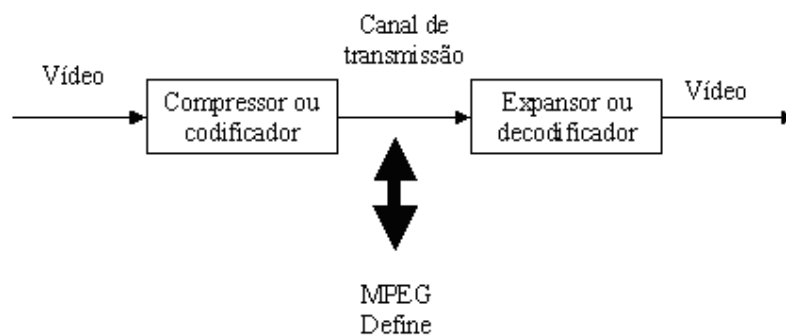


Figura 3.5: Sistema de Codificação e Decodificação de um vídeo no formato MPEG.

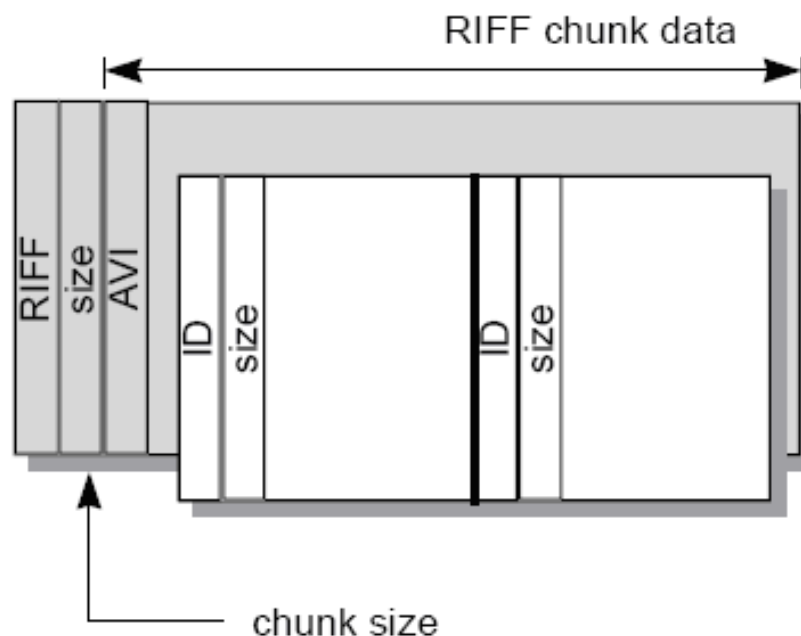


Figura 3.6: Sistema de armazenamento AVI.

O AVI (Audio Video Interleave) é um padrão criado pela Microsoft em 1992, o qual é derivado do padrão RIFF que divide os dados em blocos. Ao contrário dos outros padrões

o AVI não possui compressão.

3.3 Sistemas de Cor

Com a grande conclusão de Newton, no século XVII, de que a cor branca é composta por todas as outras cores, onde as três características básicas de um espectro de luz são, a matriz (definida pelo comprimento da onda), brilho (intensidade da luz fonte) e saturação (predominância da matriz). A formação do sistema de cores no processo aditivo é obtida a partir de duas fontes luminosas onde cores diferentes são projetadas em duas regiões, sendo assim, a interseção entre as cores há a formação de uma nova cor. Já o processo subtrativo é geralmente feito com a utilização de slides. Um exemplo é quando se emite uma luz branca (que possui todos os comprimentos de onda) sobre um filtro verde, o qual filtra todos os comprimentos de onda deixando passar somente o comprimento de onda relativa à cor verde, produzindo assim o verde. Dentro do sistema de cores podemos citar alguns padrões: RGB, CMYK, HSV, RYB, Highcolor, Imagem binária, SRGB, Munsell, entre outros.

3.3.1 RGB

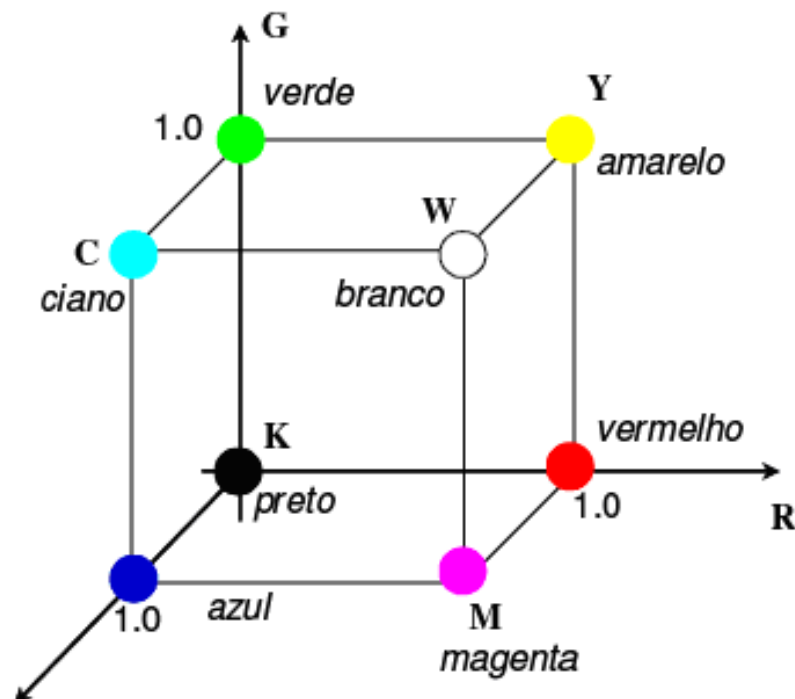


Figura 3.7: Sistema RGB.

Formado pelas cores vermelha, verde e azul. O RGB é um padrão de cores utilizado em monitores/televisores, em tubos de raios catódicos, cristal liquido ou de plasma. Utiliza

8 bits de informação para representar cada uma de suas cores (vermelho, verde, azul), podendo variar de 0 a 255. É possível alcançar mais de 16 milhões de cores com suas combinações de tons. Algumas destas cores estão representadas na Tabela 3.1:

Tabela 3.1: *Tabela com algumas das muitas combinações de cores possíveis com o sistema RGB.*

Cor	R	G	B
Preto	0	0	0
Preto Escuro	0	0	128
Azul	0	0	255
Verde Escuro	0	128	0
Turquesa	0	128	128
Azul Claro	0	128	255
Verde	0	255	0
Verde Água	0	255	128
Ciano	0	255	255
Marrom	128	0	0
Violeta	128	0	128
Azul Marinho	128	0	255
Verde Musgo	128	128	0
Cinza	128	128	128

Na Figura 3.7 - b é possível observar o modelo mapeado em um cubo. Onde o eixo X representado por valores de vermelho incrementa para a esquerda, eixo Y representado pelo azul incrementa para direita inferior e o eixo vertical Z pelo verde que incrementa para cima e a origem representa o preto.

3.3.2 CMYK

Também conhecido como modelo de cores baseado na subtração (absorção de luz), é formado pelas cores Ciano, Magenta, Amarelo e Preto, utilizado com maior frequência para impressões gráficas. A Figura ?? representa as misturas de cores de um sistema CMYK.

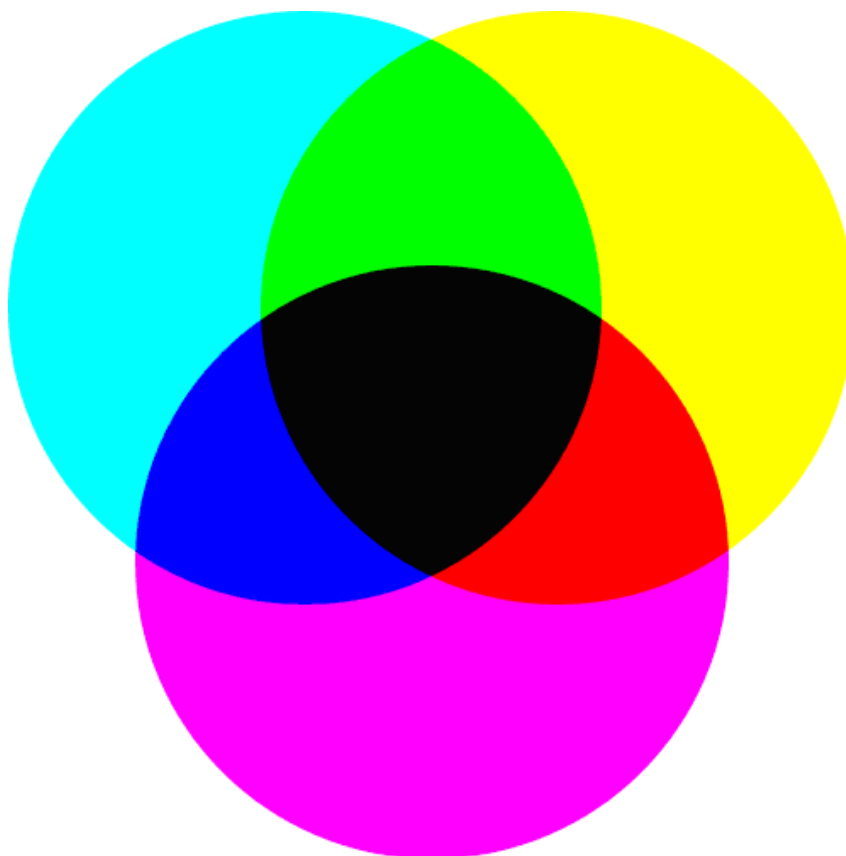


Figura 3.8: Misturas de cores no padrão CMYK.

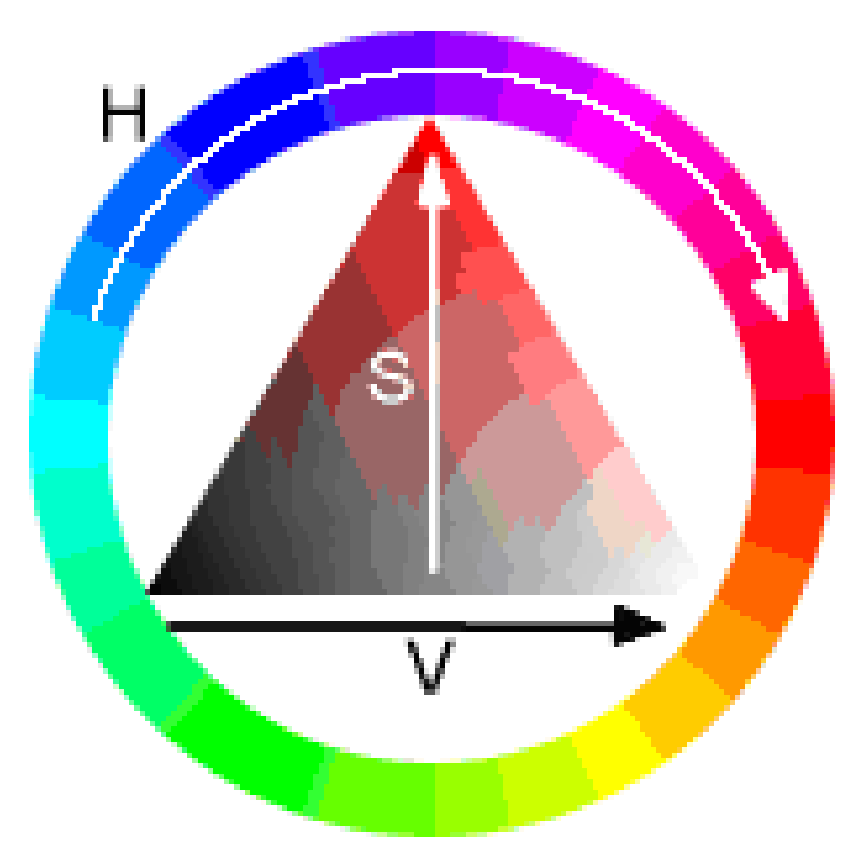


Figura 3.9: A mistura de cores representada por um círculo.

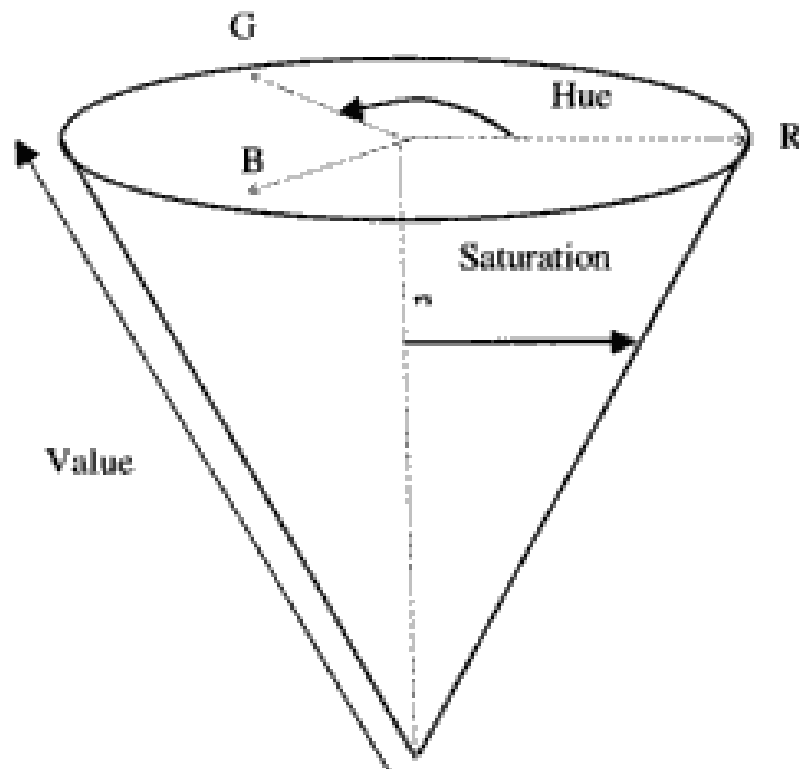


Figura 3.10: Modelo de representação do padrão HSV.

3.3.3 HSC

Também conhecido como HSB, este sistema de cores é o mais próximo à percepção da visão humana, o que leva vantagens em relação ao sistema RGB e CMYK. Este sistema é composto pela Tonalidade, Saturação e Valor, onde a tonalidade é a cor definida pela faixa de valores de 0 a 360 graus, representada por um círculo. A saturação varia pelo percentual da presença de tons de cinza de 0 a 100.

Ângulo Cor 0-60 Vermelho 60-120 Amarelo 120-180 Verde 180-240 Ciano 240-300 Azul 300-360 Magenta

?? - Seleção de nível de cores.

3.3.4 Highcolor

No Highcolor cada pixel de cor é apresentado por 2 bytes, uma cor geralmente é representada por 16 bits, mas existem alguns processadores gráficos que armazenam somente 15 bits de informação, neste caso 5 bits são selecionados para definição de uma cor (ilustrada na Tabela 1), onde $2^5 = 32$, possibilitando aproximadamente 32 mil cores por pixel.

Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 Valor Vermelho Verde Azul

Tabela2 - Sistema de cores 15bits no Highcolor Quando os 16 bits são utilizados é

possível alcançar aproximadamente 64 mil cores.