

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE – UFCG
CENTRO DE ENGENHARIA E INFORMÁTICA
UNIDADE ACADÊMICA DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**ALGORITMO PARA EXTRAÇÃO DE IMAGENS DE
FUNDO NÃO HOMOGÊNEOS USANDO O ESPAÇO DE
CORES YCBCR**

ROBERTO HIGINO PEREIRA DA SILVA

Campina Grande – Paraíba
Maio de 2006

ROBERTO HIGINO PEREIRA DA SILVA

ALGORITMO PARA EXTRAÇÃO DE IMAGENS DE
FUNDO NÃO HOMOGÊNEOS USANDO O ESPAÇO DE
CORES YCBCR

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, em cumprimento às exigências para obtenção do grau de Mestre em Ciências no domínio de Engenharia Elétrica.

Orientadores: Prof. Dr. Raimundo Carlos Silvério Freire

Prof. Dr. Hiran de Melo

Campina Grande – Paraíba
Maio de 2006

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

S586i Silva, Roberto Higino Pereira da
2006 Algoritmo para Extração de Imagens de Fundos não-Homogêneos Usando o Espaço de Cores YCbCr / Roberto Higino Pereira da Silva. Campina Grande, 2006.
105f. : il.

Referências.

Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal de Campina Grande, Centro de Engenharia e Informática.

Orientadores: Prof. Dr. Raimundo Carlos Silvério Freire e Prof. Dr. Hiran de Melo.

1 Processamento de Imagens I Título

CDU 004.383.5:621.3

**ALGORÍTMO PARA EXTRAÇÃO DE IMAGENS DE FUNDOS NÃO
HOMOGÊNEOS USANDO O ESPAÇO DE CORES YCBCR**

ROBERTO HIGINO PEREIRA DA SILVA

Dissertação Aprovada em 29.05.2006

RAIMUNDO CARLOS S. FREIRE

RAIMUNDO CARLOS SILVÉRIO FREIRE, Dr., UFCG

Orientador

HIRAN DE MELO

HIRAN DE MELO, Dr., UFCG

Orientador

FERNANDO RANGEL DE SOUZA

FERNANDO RANGEL DE SOUZA, Dr., UFRN

Componente da Banca

GLAUCO FONTGALLAND

GLAUCO FONTGALLAND, Dr., UFCG

Componente da Banca

CAMPINA GRANDE - PB
MAIO - 2006

DEDICATÓRIA

Dedico com muito carinho a minha mãezinha
Maria Pereira da Silva, meu irmão e minhas
irmãs, a minha família Nice e Rodrigo.

AGRADECIMENTOS

- A DEUS, senhor de todas as coisas.
- Aos amigos do curso de Pós-Graduação de Manaus. Pelo companheirismo e incentivo no decorrer do curso.
- Ao amigo André Luiz Printes, pelo incentivo e ajuda.
- Ao prof. Raimundo Carlos Silvério Freire, pelo apoio e acreditar nas pessoas do Norte.
- A EST/UEA, em especial ao Prof. Dr. Vicente Nogueira, pelo apoio incondicional.
- A todos os professores do curso de mestrado que me honraram com os seus conhecimentos.
- Aos amigos do laboratório, em especial ao Francisco e Will pelo incentivo e ajuda.
- Ao amigo do ITA, Fernando Nicodemos pela ajuda dispensada.
- A FAPEAM, pela ajuda financeira.
- A todos que de alguma forma me ajudaram na conclusão deste trabalho.

RESUMO

A extração de objetos em uma imagem tem várias aplicações na área da automação, tais como: reconhecimento de padrões em sistemas de vigilância, visão de robôs e outros.

Este trabalho apresenta um algoritmo estatístico de extração de imagens no espaço de cores RGB implementado em uma plataforma DSP e a análise dos resultados obtidos.

É proposto um outro método estatístico para extração de imagens em fundo desconhecido, podendo ser homogêneo ou não-homogêneo. O algoritmo proposto tem como base o espaço de cores YCbCr, é destinado a aplicações que exigem performance e tolerância a um determinado intervalo de erro. Utiliza a métrica dos máximos para determinar a distância entre os vetores desse espaço, sendo capaz de suportar pequenas variações de luminosidade. Apresenta-se a simulação que validou a funcionalidade desse algoritmo.

ABSTRACT

The images objects embedded extraction has several applications in the automation area, such as: patterns recognition in surveillance systems, robots vision and others.

An image statistical extraction algorithm in the RGB colors space was implemented in a DSP platform and the obtained results analysis are presented in this dissertation.

We propose a new statistical method for objects extraction in an unknown background (non-homogenous or homogenous), using the YCbCr space. It uses maximum metric to determine its distances between the space vectors, being capable to support small global variations and places of brightness. They are presented the simulation results that validate the algorithm functionality for applications that demand performance and be tolerant to a determined error interval.

SUMÁRIO

INTRODUÇÃO.....	12
1 TEORIA DAS CORES.....	17
1.1 Cor.....	18
1.2 Luz Visível.....	18
1.3 O Sistema Visual Humano.....	19
1.3.1 Fisiologia da Visão.....	19
1.4 Características das Cores.....	22
1.5 Modelos de Cores.....	23
1.5.1 Modelo CIE	23
1.5.2 Modelo RGB.....	25
1.5.3 Modelos YIQ e YCbCr.....	27
2. MODELOS PARA EXTRAÇÃO DE OBJETOS.....	31
2.1 Introdução.....	32
2.2 Características dos Modelos.....	32
2.3 Subtração de Imagem de Fundo.....	33
2.4 Requisitos Necessários para Extração de Objeto.....	34
2.4.1 Algoritmos de Wu & Jeng e Butler.....	34
2.4.2 Algoritmos de Elgammal e Hong & Woo.....	35
2.4.3 Robustez e Eficiência.....	35
2.4.4 Derivada Temporal.....	36
2.4.5 Correlação Normalizada entre Blocos.....	36
2.4.6 Algoritmo de Horpraset.....	37
2.5 Características Físicas das Câmeras de Vídeos.....	38
2.6 O Método Escolhido.....	38
2.7 Uma Visão do Algoritmo de Horpraset.....	39
3. ALGORITMO DE HORPRASET E PLATAFORMA DSP.....	42
3.1 Introdução.....	43
3.2 Modelo Computacional.....	43
3.3 Modelagem do Fundo.....	45
3.4 Extração de Objeto.....	47
3.5 Hardware Utilizado.....	48
3.5.1 Arquitetura Blackfin.....	50
3.5.2 Ambiente de Desenvolvimento.....	51
4. EXTRAÇÃO DE IMAGEM UTILIZANDO O ESPAÇO CORES YCbCr.....	53
4.1 Introdução.....	54
4.2 Representação do Espaço de Cores YCbCr.....	54
4.3 Algoritmo Proposto.....	56
4.4 Aquisição das Imagens.....	58
4.5 Modelagem do Fundo.....	58
4.6 Definição da Métrica Utilizada.....	59

4.6.1 Métrica do Máximo.....	59
4.6.2 Métrica da Soma.....	60
4.7 Extração de Objeto.....	60
4.7.1 Fundo Original.	61
4.7.2 Objeto..	63
4.7.3 Sombra ou Iluminação.	64
4.7.4 Outros Casos..	65
4.8 Abordagem Estatística.....	65
4.6.1 Desbalanceamento das Cores.	65
5. RESULTADOS OBTIDOS.....	68
5.1 Introdução.....	69
5.2 Simulação do Algoritmo no Matlab.....	69
5.2.1 Resultados Obtidos na Simulação.....	71
5.3 Sistema de Extração de Objetos Implementado em DSP.....	73
5.4 Algoritmo Proposto em YCbCr.....	74
CONCLUSÃO E SUGESTÕES.....	79
6.1 Análise dos Resultados.....	80
6.2 Trabalhos Futuros.....	80
RERÊNCIAS E OBRAS CONSULTADAS.....	82
APÊNDICE.....	86
ANEXO.....	90

LISTA DE FIGURAS

Figura 1.2.1: Espectro eletromagnético da banda correspondente à luz visível.....	19
Figura 1.3.2: Diagrama simplificado de uma seção do olho humano.....	20
Figura 1.3.3: Sensibilidade dos cones da retina em função do comprimento de onda.....	21
Figura 1.5.4: Funções CMF x, y e z.....	24
Figura 1.5.5: Espaço de cor CIE.	25
Figura 1.5.6: Espaço de cor RGB.....	26
Figura 1.5.7: Espaço de cor YIQ e YCbCr.....	29
Figura 2.4.8: Extração de objeto utilizando o algoritmo de Elgammal, utilizando um fundo não homogêneo.....	35
Figura 2.7.9: Algoritmo de extração de fundo: Etapa de modelagem de fundo.....	40
Figura 2.7.10: Algoritmo de extração de fundo: Etapa de extração de objeto.....	41
Figura 3.2.11: Modelo de cor no espaço tridimensional RGB.....	44
Figura 3.5.12: Arquitetura do ADSP-BF533 EZ-KIT Lite.....	49
Figura 3.5.13: Core do DSP Blackfin.....	51
Figura 3.5.14: Ambiente de desenvolvimento VisualDSP++ 4.0.....	52
Figura 4.2.15: Cubo unitário do espaço de cores RGB.....	55
Figura 4.2.16: Cubo unitário do espaço de cores YCbCr.....	55
Figura 4.3.17: Espaço de cores YCbCr.....	57
Figura 4.7.16: Fundo original, quando $Z=X=y=0$	63

Figura 4.7.19: Pixel definido como objeto, quando $Z \neq 0$, $X \neq 0$ e $y=0$	64
Figura 4.7.20: Pixel definido como sombra ou iluminação, quando $Z \neq 0$, $X=0$ e $y=Z$	65
Figura 5.2.21: Diagrama do algoritmo implementado no MATLAB.....	70
Figura 5.2.22: Resultado da simulação com MATLAB 7.....	72
Figura 5.2.23: Resultado da simulação com MATLAB 7.	72
Figura 5.2.24: Resultado da simulação com MATLAB 7.....	73
Figura 5.2.25: Imagem de um fundo, capturado pela câmera.....	75
Figura 5.2.26: Imagem do fundo com objeto, capturado pela câmera.....	75
Figura 5.2.27: Extração do objeto visualizado pela TV.....	75
Figura 5.4.28: Diagrama do algoritmo proposto implementado no MATLAB.....	76
Figura 5.4.29: Diagrama do algoritmo proposto implementado no MATLAB.....	77

LISTA DE SÍMBOLOS

Símbolo	Descrição
i	Posição (x,y) qualquer de um <i>pixel</i> em um quadro.
E_i	Valor esperado para o <i>pixel</i> i na imagem de fundo. É formado por $\mu_{i(R)}, \mu_{i(G)}, \mu_{i(B)}$
$\mu_{i(R)}, \mu_{i(G)}, \mu_{i(B)}$	Valor médio de cada uma das componentes R,G e B do <i>pixel</i> i na imagem de fundo, calculado em um número N de quadros.
I_i	Valor do <i>pixel</i> i na imagem do primeiro plano. É formado por $I_{i(R)}, I_{i(G)}, I_{i(B)}$
α_i	Distorção de Brilho.
$I_{i(R)}, I_{i(G)}, I_{i(B)}$	Cada uma das componentes R, G e B do <i>pixel</i> i na imagem do primeiro plano.
CD_i	Distorção de cor
S_i	Desvio padrão dos valores de cada um dos <i>pixels</i> de uma imagem estática de fundo. É formado por $\sigma_{i(R)}, \sigma_{i(G)}, \sigma_{i(B)}$
$\sigma_{i(R)}, \sigma_{i(G)}, \sigma_{i(B)}$	Desvio padrão de cada uma das componentes R, G e B de um <i>pixel</i> i na imagem de fundo, calculado em um número N de quadros.
a_i	Variação da distorção de brilho, fator de normalização.
b_i	Variação da distorção de cor, fator de normalização.
N	Quadros da imagem de referência
Z	Representa a distância entre o vetor E_i e I_i
X	A distância da projeção dos vetores E_i e I_i no plano $C_b C_r$
y	A distância entre os pontos da projeção dos vetores E_i e I_i no eixo da luminância Y
Y	Luminância
$P_i(\eta)$	Representa o valor do pixel i ,
$W_i(\eta)$	Valor médio do pixel i
η	Y, Cb, Cr
$\ E_i - I_i\ _s$	A métrica da soma
$\ E_i - I_i\ _M$	A métrica do máximo
$\ E_i - I_i\ $	A métrica Euclidiana

LISTA DE ACRÔNIMOS

CCD	- Charge Coupled Device
CIE	- Commission Internationale de l'Éclairage
CMF	- Colour Matching Functions
DSP	- Digital Signal Processor
FPGA	- Field Programmable Gate Array
JPEG	- Joint Photographic Experts Group
MAC	- Multiply ACcumulate
NTSC	- National Television Standards Committee
PAL	- Phase Alternate Lines
PPI	- Parallel Peripheral Interface
RGB	- Red, Green, Blue
SECAM	- Systeme Electronique Couleur Avec Memoire

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

S586i Silva, Roberto Higino Pereira da
2006 Algoritmo para Extração de Imagens de Fundos não-Homogêneos Usando o Espaço de Cores YCbCr / Roberto Higino Pereira da Silva. — Campina Grande, 2006.
105f. : il.

Referências.

Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal de Campina Grande, Centro de Engenharia e Informática.

Orientadores: Prof. Dr. Raimundo Carlos Silvério Freire e Prof. Dr. Hiran de Melo.

1— Processamento de Imagens I— Título

CDU 004.383.5:621.3

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.

INTRODUÇÃO

Nesta introdução, é apresentado o estado da arte em extração de objeto em um fundo homogêneo e não homogêneo. Também são apresentadas a motivação e a justificativa de se utilizar a extração de objetos com Processadores Digitais de Sinais, e, ao final, são apresentados os conteúdos de cada capítulo.

Nos últimos anos, a utilização de Processadores Digitais de Sinais – DSP (*Digital Signal Processors*) tem sido bastante difundida na área de engenharia elétrica. Os DSP's são amplamente utilizados no processamento digital de imagens. Várias aplicações têm sido implementadas com esses processadores. Dentre elas, destaca-se a extração de objetos a partir de um fundo homogêneo ou não homogêneo.

Os DSP's possuem uma arquitetura especializada que leva em conta os tipos de aplicações, de dados e operações aritméticas que são mais comumente utilizados. Para realizar essas operações de forma eficientes, eles possuem uma estrutura de memória, barramento e instruções bastante eficiente, tornando seu desempenho muito superior ao dos microprocessadores de uso geral.

O uso dos DSP's no processamento digital de imagens proporcionou uma grande revolução nessa área, permitindo o desenvolvimento de equipamentos com características, recursos e qualidade nunca antes possíveis, além de conseguir uma significativa redução no peso e custo final dos produtos [5].

Nas aplicações práticas que envolvem requisitos de processamento em tempo real e grande quantidade de dados, o uso dos métodos de extração de imagens por *software* torna o processamento bastante lento. Nesses casos, é imprescindível que tais métodos sejam implementados em *hardware*, tal como ocorre com a utilização de plataforma FPGA[16,17].

A extração de objeto a partir de um fundo estático ou móvel tem sido utilizada em várias aplicações que necessitam isolar objetos para análise e observação [5], tais como: os sistemas de vigilâncias de estradas [2], análise de imagens médicas [18], reconhecimentos de pessoas em ambientes vigiados [11] e no mercado de entretenimento.

Basicamente, os métodos de extração de objetos consistem em distinguir um objeto pertencente a uma imagem de primeiro plano, do fundo previamente analisado e isolá-lo para análise e manipulação.

Vários algoritmos de extração de imagens, a partir de um fundo com características próprias, vêm sendo desenvolvidos e implementados para os mais variados tipos de aplicações. Eles sempre apresentam algum tipo de dificuldade no reconhecimento de pequenas variações de iluminação generalizadas ou localizadas, que são inerentes a ambientes não controlados. Várias técnicas têm sido propostas para solucionar esse problema [23]. Nesta dissertação, segue-se uma linha de pesquisa iniciada em 2001 por Printes [17] e Parente [16], ex-alunos de pós-graduação da UFCG.

Printes [17] adaptou um método estatístico para extração de objetos de um fundo não homogêneo e validou o método utilizando um *software* de simulação numérica. No trabalho *Circuito Integrado para Extração de fundo não homogêneo de imagens dinâmicas em tempo real*, apresentou uma plataforma em FPGA para o desenvolvimento de um sistema em *hardware* capaz de capturar uma imagem, extrair o objeto principal de um fundo não homogêneo qualquer, porém estático, e inseri-lo em uma segunda imagem de fundo diferente, oriunda de outra fonte de vídeo em tempo real.

Parente [16] implementou em *hardware* o método de extração de objetos de um fundo não homogêneo, utilizando uma plataforma FPGA desenvolvida para este propósito, concluindo, assim, a etapa de extração do objeto do sistema, conforme foi proposto por Printes [17].

O algoritmo adotado por Parente [16] e Printes [17] faz uso do espaço de cores RGB, sendo necessário o cálculo do parâmetro de luminância para a extração do objeto. Para implementação deste trabalho, o espaço de cores utilizado foi YCbCr, pois baseia-se em um sinal de luminância e dois sinais de crominância.

Nesta dissertação, implementa-se em DSP o método de extração de objetos de um fundo não homogêneo, utilizando uma plataforma de desenvolvimento DSP de baixo custo e que seja capaz de capturar uma imagem através de uma câmera de vídeo colorida

convencional, extrair o objeto principal de um fundo qualquer não homogêneo, mas relativamente fixo e conhecido, para, em seguida, inserir este objeto sincronizado em uma segunda imagem de fundo diferente, proveniente de outra fonte de vídeo qualquer, em tempo real.

O algoritmo utilizado neste trabalho tem como vantagem a baixa complexidade e a exatidão dos cálculos na extração do objeto, permitindo operar em alta velocidade, devido a sua simplicidade computacional. O algoritmo começa construindo um modelo não-determinístico do fundo, usando uma abordagem estatística para, posteriormente, fazer a extração do objeto do plano de fundo [8].

O algoritmo utilizado não opera com mudanças nas cenas de fundo, porém ele é muito robusto para cenas com mudanças de iluminação e para eliminação das sombras dos objetos extraídos. Descreve-se, neste trabalho, o processo de desenvolvimento do sistema implementado numa plataforma DSP, conforme as características do algoritmo escolhido, bem como as alterações efetuadas em função da mudança do espaço tridimensional adotado.

Vale ressaltar que os trabalhos que utilizam extração de objetos em tempo real geralmente operam no espaço RGB [16,17,20,22]. Esta proposta faz uso de uma plataforma baseada em DSP para a extração do objeto, cujo *decoder* de vídeo utiliza o sistema de cores YCbCr, fazendo-se necessário operar algumas mudanças no algoritmo original [8], de modo que a imagem possa ser amostrada em tempo real.

Dentre as várias opções disponíveis no mercado, foi adotada a plataforma DSP da *Analog Device* para a implementação desse método, pois seu custo, em relação a outros equipamentos de igual desempenho, é o mais baixo.

Além da introdução, conclusão e referências, o texto desta dissertação foi dividido em cinco capítulos, um anexo e um apêndice, assim organizados.

No capítulo 1, conceituam-se vários termos utilizados em teoria das cores, de modo que os conceitos utilizados no texto desse trabalho possam ser compreendidos, bem como definem-se os espaços de cores utilizados.

No capítulo 2, é feita uma descrição da revisão bibliográfica em que são mostradas as principais técnicas e métodos atualmente utilizados para a extração de objetos de uma imagem, enfatizando-se as principais vantagens e desvantagens de cada um em relação às demais.

No capítulo 3, são apresentados os fundamentos do método proposto para extração de objetos e as adaptações necessárias para implementação desse algoritmo na plataforma DSP, bem como suas características e partes utilizadas.

No capítulo 4, é proposto um novo método de extração de imagens utilizando-se espaço de cor YCbCr. São definidos o espaço vetorial e as equações relacionadas a esse algoritmo.

No capítulo 5, apresentam-se os resultados obtidos do algoritmo de Horpraset em simulação e na plataforma DSP. É mostrado também em simulação a extração de objetos utilizando o algoritmo proposto em YCbCr.

Na conclusão e sugestões, analisam-se os resultados obtidos, fazem-se as considerações finais e propõem-se tópicos para trabalhos futuros.

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.

1

TEORIA DAS CORES

Este capítulo tem como referência básica o artigo *Teoria das cores* [13]. Definem-se alguns conceitos e características da cor, bem como algumas particularidades da visão humana. É apresentada uma visão geral sobre os modelos dos espaços de cores. No final, determinam-se os espaços de cores utilizados neste trabalho.

1.1 COR

A cor é um fenômeno físico gerado pela luz. Transmite informação sobre as condições de iluminação e a forma dos objetos visualizados, acentuando os seus contornos [13].

Quando se estuda a cor, verificam-se várias características relacionadas a ela, tais como: matiz, saturação, luminosidade e brilho. Matiz é a cor pura em sua máxima intensidade, percebida pelo observador (vermelho, amarelo, azul, etc). Saturação é o quanto a cor pura (o matiz) é diluída pela luz. Luminosidade é a intensidade da luz refletida pela superfície dos objetos. Brilho é a quantidade de luz emitida pela superfície [13]. A percepção da cor só é possível devido à existência da luz. Faz-se necessário compreender o que é a luz e como ela é percebida e interpretada.

1.2 LUZ VISÍVEL

O espectro de radiação eletromagnética (figura 1.2.1) ocupa uma grande faixa de frequências (ou comprimentos de onda) que se estende desde as baixas frequências das ondas de rádio até às altas frequências correspondentes aos raios gama. O espectro eletromagnético é habitualmente dividido em bandas segundo os efeitos ou o tipo de utilização de cada uma delas. O olho humano são sensíveis à radiação eletromagnética numa banda estreita, de comprimento de onda denominada espectro visível, o olho interpreta a luz como radiação eletromagnética. O olho humano reconhece o espectro visível dentro de um limite, que pode variar entre as pessoas [11].

A cor é essencialmente um conceito subjetivo, próprio do ser humano e consiste na interpretação que o sistema sensorial e o cérebro dão aos diferentes comprimentos de ondas. A luz recebida pelos olhos gera estímulos nervosos, provocados pela absorção dos fótons da radiação eletromagnética, com comprimento de onda compreendido entre 400 e 700 nm (figura 1.2.1)

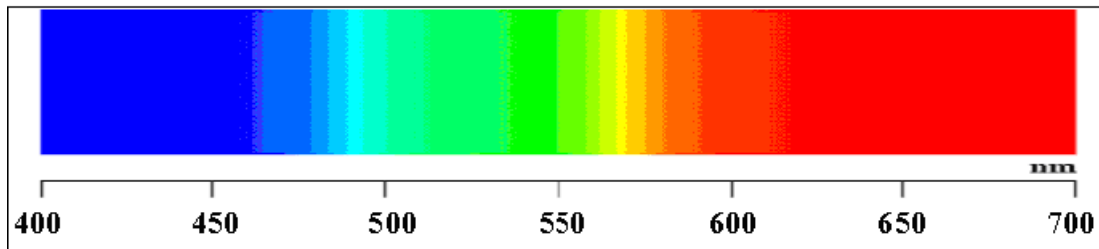


Figura 1.2.1: Espectro eletromagnético da banda correspondente à luz visível.

Fonte: GONZALES, Raphael C., WOODS, Richard C. *Processamento Digital de Imagens*.

1.3 O SISTEMA VISUAL HUMANO

É através dos olhos que se consegue perceber as cores e a luz, de modo que é importante apresentar algumas informações básicas do processo de percepção visual. Apresenta-se a seguir um breve resumo dos mecanismos do sistema visual humano.

1.3.1 Fisiologia da visão

Muitos equipamentos que trabalham com imagens tiveram a sua concepção a partir do sistema visual humano [5], pelo menos na recepção e detecção da luz.

No olho humano, quando a luz converge na retina, forma-se a imagem. Nela existem células sensíveis à luz, chamadas de bastonetes e cones, as quais transformam a luz absorvida em impulsos nervosos, que são comunicados ao cérebro pelo nervo óptico (figura 1.3.2). O cérebro recebe estes impulsos e processa-os, interpretando-os como imagens. Quando persistirem durante algum tempo na retina e se um número suficiente de imagens forma-se no cérebro por unidade de tempo, o cérebro interpreta as diferenças entre imagens sucessivas como informação sobre o movimento dos objetos visionados [5].

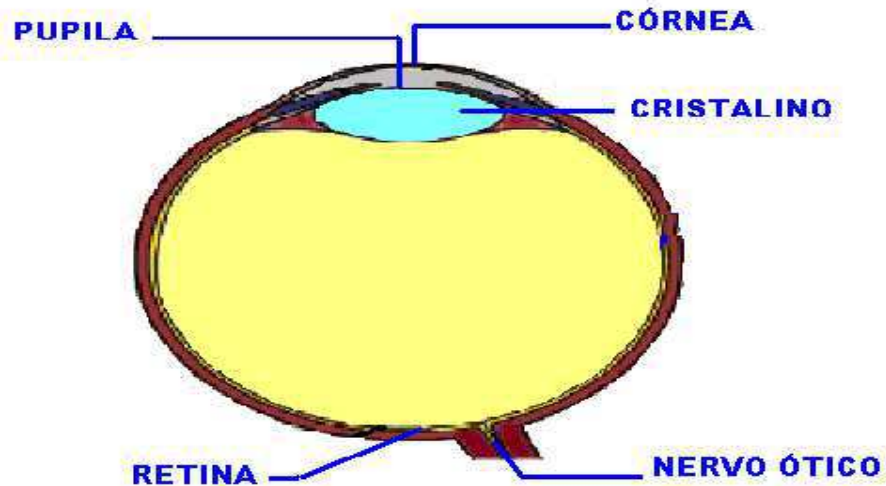


Figura 1.3.2: Diagrama simplificado de uma seção do olho humano.

O olho humano possui entre 115 e 120 milhões de bastonetes e cerca de 6,8 milhões de cones. Os bastonetes são sensíveis à intensidade luminosa em toda a gama de comprimentos de onda a que o olho humano é sensível. Eles proporcionam a detecção das formas dos objetos, que resulta na identificação das arestas pela descontinuidade da luminosidade. Sendo sensíveis à luz em todos os comprimentos de onda, os bastonetes não conseguem detectar as cores e, sim, sua luminosidade.

Os cones são células da retina com formas cônicas, mais curtas que os bastonetes. Ao contrário dos bastonetes, os cones são sensíveis à luz apenas em certas gamas de comprimento de onda. Assim, existem cones sensíveis à luz na faixa do vermelho (ρ), verde (γ) e azul (β). As gamas de comprimentos de onda a que cada tipo de cone é sensível estão parcialmente sobrepostas, conforme é mostrado na figura 1.3.3. É por este motivo que é errado falar em cones de uma determinada cor, como se cada tipo de cone fosse sensível a um comprimento de onda único e bem determinado.

Os cones necessitam de níveis de luminosidade mais elevados do que os bastonetes, e, por esta razão, o olho humano não é capaz de detectar a cor dos objetos em condições de iluminação muito fraca como à noite.

A distribuição de cones e bastonetes na retina não é uniforme, de modo que a visão foi dividida em central e periférica. A visão central permite identificar detalhes, cores e realizar a focagem do olho. A visão periférica detecta formas e movimentos de objetos sem grandes detalhes e sem cor.

A informação de cor produzida pelos cones não é suficiente para a detecção das arestas e formas dos objetos, mas pode reforçar positivamente a informação proveniente dos bastonetes se existirem diferenças de cor em torno das arestas. No entanto, em situações em que a diferença de cor entre objetos não é facilmente perceptível, a informação de cor proveniente dos cones tem o efeito de dificultar a percepção das arestas e formas dos objetos.

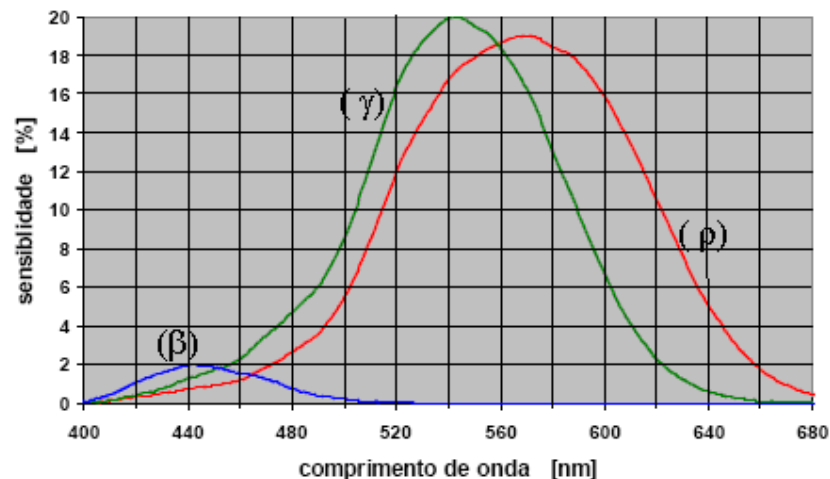


Figura 1.3.3: Sensibilidade dos cones da retina em função do comprimento de onda.

Vale ressaltar que cada tipo de cone é sensível à luz num determinado comprimento de onda. Os cones da retina detectam a luz na gama de comprimentos de onda situada entre 400 e 680 nm, podendo variar de pessoa para pessoa. Portanto, é possível que uma pessoa possa enxergar uma determinada cor ligeiramente diferentemente de outras pessoas.

A sensibilidade do olho humano não é uniforme, existindo, portanto, um comprimento de onda para o qual a sensibilidade de cada tipo de cone atinge seu máximo, e é praticamente nula para comprimentos de onda nos limites inferior e superior, como pode ser verificado na figura 1.3.3. Portanto, os comprimentos de onda inferiores a 400 nm são detectados apenas pelos cones de tipo β , os comprimentos de onda superiores a 680 nm são detectados apenas pelos cones de tipo ρ e comprimentos de onda compreendidos entre 400 e 680 nm são detectados pelos três tipos de cones, mas cada tipo de cone detecta um dado comprimento de onda com uma sensibilidade diferente, o que dá origem a impulsos nervosos de diferentes intensidades. Para exemplificar essa ponderação, pega-se a cor amarela que corresponde ao comprimento de onda de 550 nm e os pesos da sensibilidade de β , γ , e ρ são de 0%, 99% e 80%, respectivamente. É a diferença entre as respostas dos três tipos de cones que permite interpretar diferentes comprimentos de onda como correspondendo a cores diferentes. Esta interpretação é conhecida como *teoria dos três estímulos* [13]. Assim, qualquer cor do espectro visível pode ser reproduzida pela adição dos resultados obtidos pelo estímulo dos três tipos de cones de forma diferente. A cor percebida depende unicamente da relação entre os pesos de β , γ , e ρ .

1.4 CARACTERÍSTICAS DAS CORES

O modelamento da cor é extremamente importante no estudo de processamento digital de imagens, pois a apresentação das imagens necessita de informações sobre a cor dos objetos nas cenas amostradas. Esses modelos devem levar em conta o formato das informações, que deve ser coerente e precisa, introduzida com um menor número de erro possível. Tendo em vista a teoria dos três estímulos [13], é intuitivo que qualquer modelo de cor deva possuir três parâmetros. O principal problema desta modelação de cor é definir que grandezas devem ser associadas a cada um dos três parâmetros (β , γ , e ρ). Essa escolha é influenciada pela natureza dos dispositivos de saída, tanto que, em função disso, foram criados vários sistemas de cores, mostrados até o final deste capítulo.

A luz emitida conforme um comprimento de onda concisa produz uma cor pura no espectro visível. Se a energia emitida pela fonte for baixa, a cor aparecerá como um sombreado. Se a energia for alta, a cor é percebida como uma cor viva e brilhante, e, se na

cor emitida for acrescentada uma luz branca e uniforme em todos os comprimentos de onda, a cor ficará mais clara. Portanto, essas características influenciaram na criação dos modelos de cores e na sua evolução.

1.5 MODELOS DE CORES

O propósito de um modelo de cores é facilitar a especificação das cores em alguma forma padrão. Essencialmente, um modelo de cor é uma especificação de um sistema de coordenadas tridimensionais e de um subespaço. Dentro deste sistema, cada cor é representada por um único ponto [5]. A maioria dos modelos de cores em uso é direcionada a dispositivos físicos, tais como monitores e impressoras coloridas.

1.5.1 Modelo CIE

A teoria dos três estímulos [13] sugere que basta emitir três cores primárias detectáveis pelos cones da retina para que se possam reproduzir todas as cores visíveis, variando somente a proporção das quantidades de luz emitida por cada uma das fontes primárias. Estas proporções seriam determinadas pelas curvas de resposta característica de cada um dos três tipos de cones.

Na prática, as três cores primárias emitidas por cada um dos canhões do tubo de raios catódicos de um monitor colorido (vermelho a 700 nm, verde a 546 nm e azul a 436 nm) não correspondem às cores detectadas pelos cones. Há então a necessidade de modificar as funções peso aplicadas a cada uma das componentes primárias emitidas.

Devido à incapacidade de modelos baseados na mistura de cores (vermelhas, verdes e azuis), de representar todas as cores do espectro visível, isto levou a CIE (Commission Internationale de l'Éclairage) a criar, em 1931, um modelo com três cores primárias denominadas X, Y e Z, que substituiu as cores primárias vermelha, verde e azul, e que possuíam funções peso intrinsecamente positivas que passasse a ser possível representar todas as cores do espectro visível. Estas funções x_γ , y_γ e z_γ foram chamadas de *funções combinadas de cor* ou CMF (*Colour Matching Functions*). O gráfico que representa as

funções é mostrado na figura 1.5.4. Foram calculadas a partir das funções de mistura para as cores vermelha (r), verde (g) e azul (b).

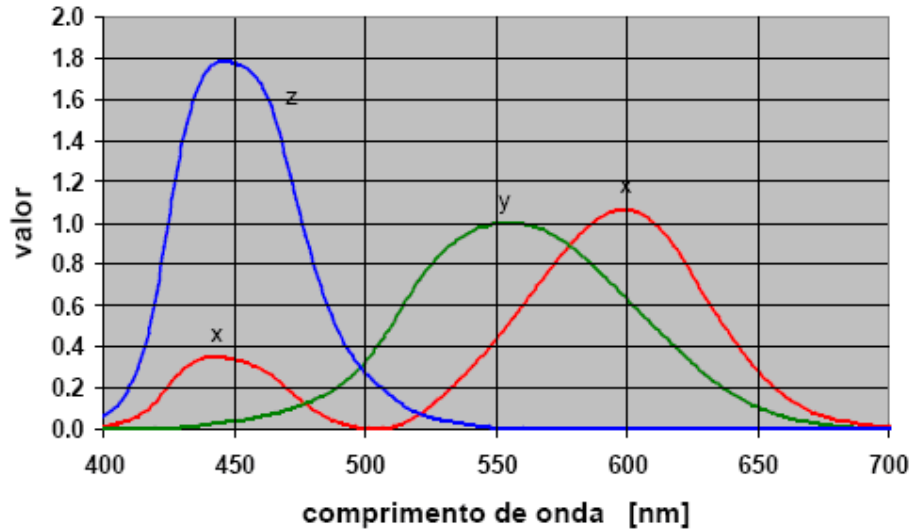


Figura 1.5.4: Funções CMF x, y e z.

A relação entre as cores primárias CIE e as cores r, g e b são dadas por:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0,49 & 0,31 & 0,20 \\ 0,17697 & 0,8124 & 0,01063 \\ 0 & 0,01 & 0,99 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} \quad (2,1)$$

em que

$$x = 0,49r + 0,31g + 0,20b \quad (2,2)$$

$$y = 0,17697r + 0,81g + 0,01063b \quad (2,3)$$

$$z = 0,01g + 0,99b \quad (2,4)$$

Pela transformação inversa destas equações, pode-se obter os valores de luzes vermelha (700 nm), verde (546 nm) e azul (436 nm) necessárias para reproduzir qualquer

cor do espectro visível a partir das cores primárias CIE. Na figura 1.5.5, observa-se o espaço de cores do modelo CIE.

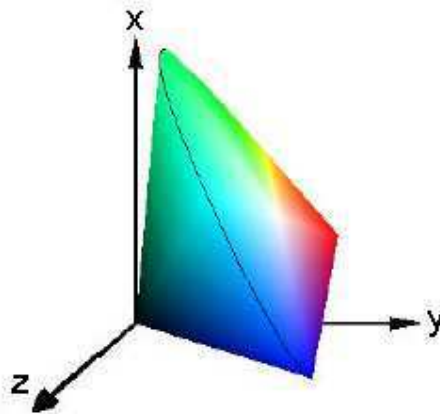


Figura 1.5.5: Espaço de cor CIE.
Fonte: LOPES, J. M. Brisson. *Teoria das cores*.

1.5.2 Modelo RGB

A maioria dos modelos atualmente em uso foi concebida para ser aplicada em um *hardware* específico (tais como os monitores coloridos e impressoras), no qual são envolvidos processos de manipulação de cores. Na prática, um dos modelos mais comumente utilizado em processamento de imagens é o RGB, cuja sigla deriva da junção das primeiras letras dos nomes das cores primárias em língua inglesa: Red, Green e Blue, traduzidas para a língua portuguesa como vermelho, verde e azul.

No modelo RGB, cada cor aparece nos seus componentes espectrais primários de vermelho, verde e azul, sendo fundamentada num sistema de coordenadas cartesianas. O subespaço de interesse é o cubo unitário em que:

$$0 \leq (R,G,B) \leq 1 \text{ (todos os valores de R,G e B estão no intervalo [0,1]).}$$

Os valores R, G e B estão nos três cantos; ciano (azul esverdeado), magenta (vermelho muito vivo, ligeiramente arroxeadado) e amarelo estão nas outras extremidades; o preto, que corresponde à intensidade mínima das cores, está na origem e o branco, que corresponde à representação simultânea das três cores primárias com intensidade máxima, está no canto mais distante da origem, conforme mostrado na figura 1.5.6. Nesse modelo, a escala de cinza estende-se do preto até o branco ao longo de uma linha reta, e as cores são pontos dentro ou sobre o cubo, definidas por vetores, estendendo-se a partir da origem.

O modelo RGB está intimamente associado às superfícies emissoras de luz. É por esta razão que ele é utilizado em muitos equipamentos que manipulam a emissão de luz, tais como os monitores e os televisores a cores.

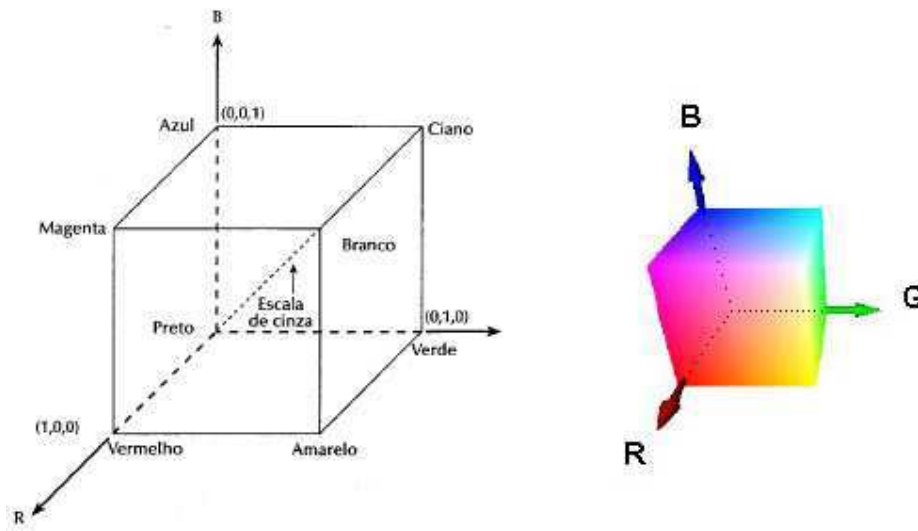


Figura 1.5.6: Espaço de cor RGB.
Fonte: LOPES, J. M. Brisson. *Teoria das cores*.

Imagens no modelo de cores RGB consistem em três planos de imagens independentes, um para cada cor primária. Quando geradas num monitor RGB, essas três imagens combinam-se sobre a tela fosfórea para produzir uma imagem de cores compostas. Portanto, o uso do modelo RGB para processamento de imagens faz sentido quando as imagens são naturalmente expressas em termos de planos de três cores.

Geralmente, as implementações do modelo RGB nos sistemas de vídeos empregam valores inteiros entre 0 e 255 para representar o valor da intensidade de cada componente em vez de valores reais normalizados entre 0 e 1, pois o processamento de valores inteiros (ponto fixo) é muito mais rápido do que o processamento de valores reais (ponto flutuante), além de que a notação com inteiros é mais simples do que a notação com valores reais fracionários. A discretização em 256 valores de intensidade é mais do que suficiente para o olho humano, dado que este consegue distinguir entre um mínimo de 16 intensidades na zona do azul e um máximo de 23 intensidades na zona do amarelo.

1.5.3 Modelos YIQ e YCbCr

Os modelos de cor YIQ e YCbCr são baseados na característica de que o olho humano é mais sensível à luz do que as cores. Esses sistemas foram criados para permitir compatibilidade entre os sistemas de televisão a cores e as televisões monocromáticas. O sistema NTSC (*National Television Standards Committee*), criado em 1953, utiliza o modelo YIQ e os sistemas SECAM (*Séquence Electronique Couleur avec Mémoire*) e PAL (*Phase Alternating Line*), criados em 1961, utilizam o modelo YCbCr, definido pela norma CCIR 601-2 [5].

No contexto de imagens digitais, o modelo YCbCr é também conhecido como YUV. Os modelos YIQ e YCbCr são muito semelhantes, pois ambos têm como base a separação dos sinais de cor RGB em um sinal de luminosidade, chamado de luminância (Y), e dois sinais de cores, conhecidos como crominância ou diferença de cores. Basicamente, os modelos YIQ e YCbCr são recodificações do modelo RGB para eficiência da transmissão e para manutenção da compatibilidade com os padrões monocromáticos de TV. Conseqüentemente, o componente Y desses sistemas fornece todas as informações de vídeo necessárias para um aparelho de televisão monocromática [5].

A principal vantagem dos modelos YIQ e YCbCr em processamento de imagens é que a luminância (Y) e as informações de cores (IQ e CbCr) são desacopladas. Vale lembrar que a luminância é proporcional à quantidade de luz percebida pelo olho humano. Assim, a importância desse desacoplamento é que a componente de luminância de uma

imagem pode ser processada sem afetar o seu conteúdo de cor, não acontecendo isso no modelo RGB.

A ponderação dos valores das componentes RGB de uma dada cor define a luminância, portanto:

$$Y = 0,299R + 0,587G + 0,144B \quad (2.5)$$

As informações de cor no modelo YIQ estão contidas nos parâmetros I e Q, sendo calculadas pelas diferenças ponderadas entre as componentes vermelha e azul da cor no espaço RGB e a luminância Y de modo que:

$$I = 0,74(R-Y) - 0,27(B-Y) \quad (2.6)$$

$$Q = 0,48(R-Y) - 0,41(B-Y) \quad (2.7)$$

Assim, obtém-se a matriz de transformação do espaço de cor RGB para o espaço YIQ:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0,299 & 0,587 & 0,144 \\ 0,596 & -0,275 & -0,321 \\ 0,212 & -0,523 & 0,311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.8)$$

Desta transformação resulta um espaço YIQ, em que Y está compreendido no intervalo [0;1]:

$$I \text{ em } [-0,596 \text{ a } +0,596]$$

$$Q \text{ em } [-0,523 \text{ a } +0,523].$$

O parâmetro Y do modelo YCbCr é idêntico ao parâmetro Y do modelo YIQ. A diferença entre os dois modelos reside na definição dos parâmetros de crominância que, no modelo YCbCr, são definidos como as diferenças entre as componentes B e R da cor no espaço RGB e a luminância Y, de modo que:

$$Cb' = B - Y \quad (2.9)$$

$$Cr' = R - Y \quad (2.10)$$

Resultando na matriz de transformação:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0,299 & 0,587 & 0,114 \\ -0,169 & -0,331 & 0,5 \\ 0,5 & -0,419 & -0,0813 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.11)$$

A norma CCIR 601-2 estabelece os intervalos de variação de Cb' e Cr' para um intervalo $[-0,5; +0,5]$. Na figura 1.5.7, são mostrados os espaços de cores dos modelos YIQ e YCbCr. Eles são paralelepipedais, em que a cor negra se encontra na origem dos espaços e a cor branca no vértice superior. Os dois modelos diferem, naturalmente, na localização das outras cores.

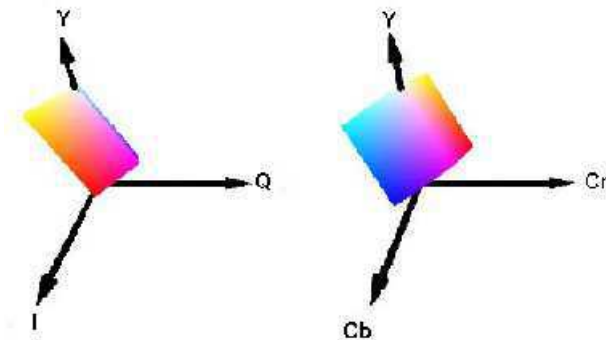


Figura 1.5.7: Espaço de cor YIQ e YCbCr .
Fonte: LOPES, J. M. Brisson. *Teoria das cores*.

Nesta dissertação utilizam-se os espaços de cores RGB e YCbCr como pilares para a construção de um algoritmo, para capturar objeto de fundo não-homogêneo. Portanto, os

conceitos discutidos neste capítulo são fundamentais para a o bom entendimento desta dissertação.

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.

2

MODELOS PARA EXTRAÇÃO DE OBJETOS

Este capítulo apresenta de forma simples e compacta um estudo sobre algoritmos para a detecção de objetos em uma cena com fundo estático. É feita uma breve explanação das principais técnicas de extração de imagem, dando ênfase à subtração de imagem de fundo. No final deste capítulo, é estabelecido um algoritmo para detectar e segmentar um objeto sobre um fundo qualquer a partir de uma seqüência de imagens em cores, capturadas por uma câmera.

2.1 INTRODUÇÃO

Muitos modelos foram propostos para extrair objetos de uma cena [2,4,7,8,10,22]. Esses modelos têm características próprias que variam conforme a aplicação a que se destinam. Usam técnicas com o objetivo de resolver alguns problemas específicos da extração de objetos [23], tais como: sombras, mudanças de iluminação, reflexões, camuflagem, etc. Em ambientes não controlados, essas cenas estão sujeitas à variação na iluminação e conseqüentemente à presença de sombras.

2.2 CARACTERÍSTICAS DOS MODELOS

Dentre os métodos de extração de objetos estudados para realizar esse trabalho, o mais simples e freqüentemente encontrado em sistemas experimentais de computação utiliza a cor ou intensidade de brilho como parâmetros de entrada e um modelo de imagem de fundo que representa os valores das características esperadas de cada *pixel* na imagem de fundo. As estatísticas da imagem de fundo são freqüentemente modeladas para cada *pixel* e armazenadas em um modelo de referência.

O modelo de referência é construído durante uma fase de aprendizado, enquanto a imagem de fundo está sem nenhum objeto da imagem do primeiro plano. Na fase de segmentação, a imagem do primeiro plano é detectada *pixel a pixel* sempre que difere significativamente da distribuição dos valores esperados para a imagem de fundo contidos no modelo de referência. Essas fases são modeladas matematicamente de modo a resolver ou atenuar alguns problemas anteriormente identificados. Observa-se que os modelos de subtração de objetos ainda apresentam dificuldades para solucionar esses problemas que ocorrem no ambiente em que o sistema está inserido.

2.3 SUBTRAÇÃO DE IMAGEM DE FUNDO

A capacidade de detectar objetos em movimento a partir de seqüências de vídeo é fundamental em muitos sistemas de visão computacional. Essa capacidade permite que os sistemas foquem a atenção nos objetos que estão em movimento e que possivelmente são uma peça crucial na execução da tarefa para a qual foram programados. Uma abordagem muito utilizada para a segmentação de objetos em movimento do fundo da cena é a subtração de fundo. A idéia da subtração de fundo é subtrair da imagem atual uma imagem de referência, essa imagem adquirida é estática, sua aquisição e processamento são feitos durante certo intervalo de tempo, conhecido como período de treinamento. Assim, o fundo pode ser qualquer um, desde que permaneça estático.

O processo de subtração de fundo de uma cena com objeto depende muito do modelo adotado para a representação deste fundo, pois esse modelo implica diretamente na qualidade dos resultados obtidos. Uma abordagem simples e muitas vezes satisfatória consiste em calcular uma imagem de referência do fundo como sendo a média de um conjunto de imagens. Posteriormente, quando um objeto entra na cena, este causa uma distorção na imagem capturada com relação à imagem de referência. Assim, os *pixels* da imagem capturada correspondentes ao objeto são aqueles nos quais ocorrem mudanças significativas.

Desde que a técnica de subtração de fundo foi reconhecida como um método para detectar objetos em movimento, apareceram na literatura diversos modelos e abordagens para a implementação destes algoritmos. Alguns deles exploram a diferença estatística e/ou probabilística de cor e luminância [8,11,14,22] entre a imagem atual e a imagem de referência (imagem de fundo), a qual é treinada durante um período de tempo ou com base em um número determinado de imagens [8,16,17]. Outros modelos têm como base outras características da imagem, como: movimento do objeto [2], diferença entre quadros adjacentes [18], correlação normalizada entre blocos [12]. Além dos métodos citados anteriormente, há outros mais complexos, tais como: utilização de vários modelos gaussianos para segmentação de imagens [6], modelo estático geométrico [9], que utiliza duas câmeras e um mapa de disparidade para extração de objetos de um fundo e o

Wallflower [23], que combina técnicas de análise de *pixel*, predição linear e atualização da imagem de fundo.

No contexto da pesquisa desta dissertação, o algoritmo de segmentação de fundo foi utilizado para segmentar o objeto de um fundo estático qualquer, sobre o qual ele estará se movimentando. Esta escolha foi devido à simplicidade de sua implementação e ao tempo utilizado na computação ser bastante reduzido, atendendo assim, os objetivos deste trabalho.

2.4 REQUISITOS NECESSÁRIOS PARA EXTRAÇÃO DE OBJETO

A maioria dos sistemas de extração de objeto processa as imagens capturadas em tempo real. Esta é uma condição deste trabalho. Como a etapa de extração de fundo é apenas a parte inicial deste sistema, a eficiência em termos de tempo de processamento é um requisito importante.

Existem problemas que tornam o processo de segmentação, via subtração de fundo, bastante difícil. Qualquer mudança física na iluminação altera a imagem do fundo, o que torna o modelo inadequado e afeta o desempenho do algoritmo. Embora muitos dos algoritmos disponíveis na literatura tenham desempenhos satisfatórios, são poucos os que conseguem lidar razoavelmente bem com as mudanças globais e locais da iluminação, tais como sombras e variações nas fontes de luz.

2.4.1 Algoritmos de Wu & Jeng e Butler

Os algoritmos de Wu & Jeng [25] e de Butler et al. [3] têm desempenho muito prejudicado pela presença de sombras. No estudo feito desses algoritmos, observa-se que as sombras são detectadas como sendo parte do objeto e não se encontra uma alternativa para destacá-las. Portanto, a morfologia do objeto fica totalmente deformada na segmentação. Esses algoritmos são adequados somente para a extração de fundos que não contenham sombras do objeto.

Observa-se, na simulação, que o próprio movimento dos objetos gera sombras e ocasiona mudanças na iluminação. Muitas vezes estas sombras são detectadas como sendo parte do objeto, afetando a eficácia do algoritmo.

2.4.2 Algoritmos de Elgammal e Hong & Woo

Existem ainda os algoritmos que tratam das sombras, mas que se tornam instáveis na presença delas. Nessa linha de pesquisa, analisam-se os algoritmos de Elgammal et al. e de Hong & Woo [7]. Foi simulado o algoritmo de Elgammal no MATLAB [15], utilizou-se uma imagem estática com fundo fixo e homogêneo e verificou-se que muitos pontos da sombra do objeto foram considerados como sendo o objeto, como é mostrado na figura 2.4.8. Foram feitas algumas inferências nos algoritmos a fim de ajustar seus limiares (valores de *threshold*). Verificou-se que esse procedimento pode melhorar esta situação, no entanto é difícil e não existe um padrão de atribuição de valores do limiar que torne fácil a tarefa de excluir os pontos da sombra na segmentação.

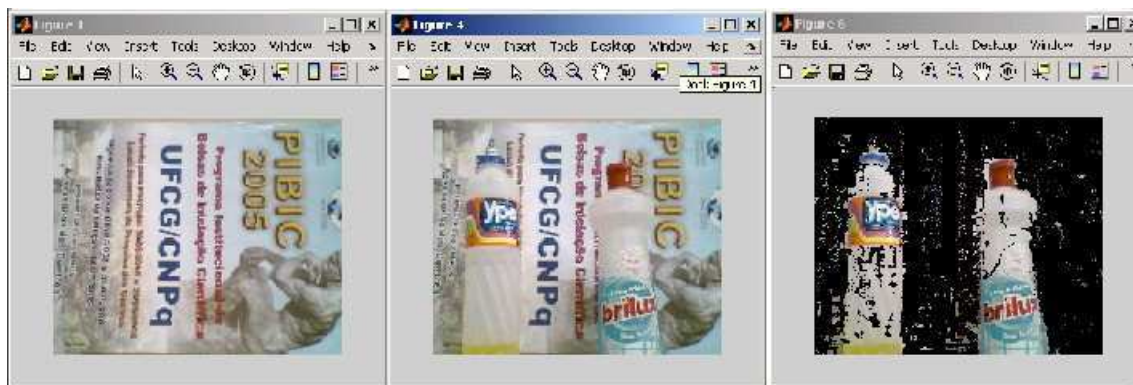


Figura 2.4.8: Extração de objeto com algoritmo de Elgammal, simulado com MATLAB.

2.4.3 Robustez e eficiência

Também é importante considerar o fator de robustez e eficiência na detecção do objeto em movimento, característica que visa permitir um alto grau de detecção e uma segmentação que represente realmente a morfologia do objeto. No contexto desta pesquisa, foi analisada a qualidade de detecção e segmentação da morfologia do objeto. O próprio movimento dos objetos gera sombras e ocasiona mudanças na iluminação. Às vezes, estas sombras são detectadas como sendo parte do objeto, afetando a eficácia do algoritmo.

No contexto desta pesquisa, as análises qualitativas foram feitas baseadas na qualidade de detecção e segmentação da morfologia do objeto.

2.4.4 Derivada temporal

Alguns algoritmos utilizam um método chamado de derivada temporal, que é utilizado em sistemas para detecção de pessoas [35]. Nesses sistemas, a segmentação também utiliza um modelo de referência da imagem de fundo adquirido em uma fase de aprendizado. Durante essa fase, cada *pixel* é representado por três valores: máximos, mínimos e a diferença máxima entre quadros consecutivos. A classificação do *pixel* é realizada comparando as novas imagens com o modelo de referência. Os *pixels*, cujas intensidades mínima ou máxima forem superiores à máxima diferença esperada, são considerados pertencentes à imagem principal. A determinação dos limiares de comparação baseada apenas em valores máximos e mínimos está sujeita a erros de detecção, visto que estes valores não constituem uma medida de dispersão muito precisa, pois não levam em consideração os valores intermediários dos *pixels*, e pequenas variações na cena ou flutuações na câmera podem induzir erros de detecção no sistema.

2.4.5 Correlação normalizada entre blocos

Algoritmos que utilizam correlação normalizada entre blocos [29] iniciam em uma fase chamada de aprendizagem, em que são adquiridos alguns quadros para obtenção da média e do desvio em relação à média. Cada imagem na fase de aprendizado é dividida em blocos, e cada bloco tem a sua média e o desvio médio dos *pixels* agrupados no bloco. A segmentação é realizada dividindo-se as novas imagens em blocos similares aos da fase de aprendizado e comparando-os com os blocos de referência. O bloco cujo valor médio estiver com um desvio maior que o esperado é classificado como imagem do primeiro plano. Este método busca uma maior exatidão na segmentação, dividindo a imagem em blocos menores. Os limiares de decisão estão baseados apenas na média e no desvio em relação à média, que ainda estão sujeitos a erros de detecção em função de variações de

iluminação e da própria câmera. Neste algoritmo, há também uma perda significativa de resolução na imagem segmentada [23], não servindo para o propósito desta dissertação.

2.4.6 Algoritmo de Horpraset

O método estatístico estabelecido por Horpraset et al. [8] visa à extração de objetos de fundos não homogêneos e desconhecidos. O modelo é baseado na diferença entre as cores medida pelas distorções de brilho e de cromaticidade, sendo bastante robusto na presença de sombras e nas variações originadas pelos sensores da câmera.

A primeira fase deste método consiste na aquisição de uma imagem de referência e do cálculo de parâmetros estáticos que expressam as características da imagem de fundo, tais como: variação de brilho e variação de cor. Na primeira etapa da fase de aquisição da imagem de fundo, são adquiridos alguns quadros utilizados para obtenção de uma imagem média. Em seguida, uma nova seqüência de imagens é adquirida para a determinação do desvio padrão dos *pixels* da imagem. Este parâmetro faz uma estimativa das variações de cada *pixel* em função das variações geradas pelos sensores da câmera. Na terceira etapa, novas imagens são adquiridas e processadas para determinação das distorções de brilho e de cor de cada *pixel* na imagem, isto é, a “diferença” de brilho e cor dos *pixels* de referência em relação aos novos *pixels* adquiridos. Em seguida, é determinada uma média de distribuição das distorções de brilho e cromaticidade que expressa a variação destas distorções. A partir destes valores são gerados histogramas para distorção de brilho e cor e automaticamente são estabelecidos os limiares de comparação. Desta forma, é finalizada a etapa de aprendizado da imagem de fundo. Para extração de objetos da imagem de fundo, são calculadas as distorções de brilho e cor para as novas imagens e comparadas com os limiares determinados na fase de aprendizado.

Levando-se em conta os aspectos analisados nos algoritmos citados, percebe-se que a sombra e a iluminação são os principais fatores que devem ser considerados na extração de um objeto em um fundo homogêneo. Existem situações para as quais é possível isolar as sombras, entretanto pequenas mudanças na iluminação tornam os limiares escolhidos inadequados, fazendo com que estes algoritmos não tenham um desempenho aceitável.

Portanto, é indispensável considerar modelos que permitam saber que houve apenas mudança na iluminação e não na cor original do fragmento da imagem. Dentro do estudo sobre algoritmos de subtração de fundo, encontram-se algoritmos que conseguem fazer esta distinção. O algoritmo de Horprasert et al. [8] é um deles, pois calcula a distorção do brilho e a distorção de cor de cada ponto no espaço RGB, tendo um desempenho aceitável para os requisitos do presente estudo.

2.5 CARACTERÍSTICAS FÍSICAS DAS CÂMERAS DE VÍDEOS

Fazem-se algumas considerações que se julgam necessárias para entender algumas características físicas que influenciam as câmeras de vídeo. Os sensores utilizados nas câmeras tipicamente transformam cores de um espaço de dimensão infinita em cores em um espaço tridimensional RGB pelos filtros de cores nas bandas vermelha, verde e azul [5]. Além da perda de precisão na digitalização das cores, existe uma contínua variação de cor ao longo do tempo.

Raramente é possível obter o mesmo valor RGB de cor num ponto específico da imagem durante certo período de tempo. Ruídos inerentes aos equipamentos e as flutuações de iluminação presentes nas fontes de luz fazem variar estes valores. Além disso, existe uma variação desigual nos filtros de cores nas bandas vermelha, verde e azul, o que indica que existe uma sensibilidade diferente para cada uma destas bandas (a variação é diferente em cada canal de cor). A visão dos seres humanos tem como uma de suas características a constância das cores. O olho humano tende a atribuir uma cor constante a um objeto, embora existam mudanças na iluminação ao longo do tempo ou espaço.

2.6 O MÉTODO ESCOLHIDO

Baseado nas análises qualitativas dos métodos pesquisados, e levando-se em consideração certas características da visão humana, escolheu-se o método estatístico de

Horprasert [8] como a técnica a ser utilizada no sistema de extração de objetos implementado neste trabalho.

O algoritmo de Horprasert et al. [8] tenta reproduzir a habilidade da constância de cores na modelagem e subtração do fundo na cena. A técnica necessita de uma fase de aprendizado em que são capturados alguns quadros para determinação de parâmetros estatísticos que são utilizados na determinação automática dos limiares de comparação adequados para segmentação da imagem. Os limiares são estabelecidos pela análise de algumas características da imagem, quais sejam: imagem média, desvio padrão, distorção de brilho e distorção de cor, que servem de base para a construção de histogramas com o comportamento do brilho e da cor na imagem de referência. Deste histograma, estabelecem-se os limiares para comparação e classificação da imagem.

2.7 UMA VISÃO DO ALGORITMO DE HORPRASET

O primeiro passo do algoritmo de Horprasert et al. [8] é capturar a imagem através de uma câmera. Ele divide-se basicamente em duas partes: modelagem do fundo (figura 2.7.9) e subtração do fundo (figura 2.7.10). Cada etapa realiza operações a fim de encontrar os parâmetros para análise dos *pixels* e conseqüentemente para a extração do objeto.

No processo de modelagem do fundo, conforme figura 2.7.9, as imagens de fundo são capturadas sem o objeto a ser extraído. Com estas imagens, são calculados os parâmetros do algoritmo, como: média dos quadros ou imagem média, desvio padrão, distorção de brilho, distorção de cor, histogramas da distorção de brilho e cor e limiares de comparação.

No processo de extração do fundo, conforme figura 2.7.10, é inserido um objeto na imagem de fundo que é capturada pela câmera e enviada para processamento da extração. Na operação de subtração de fundo, o objeto é extraído do fundo utilizando os parâmetros calculados na etapa anterior.

Portanto, o algoritmo de Horprasert et al. [8] baseia-se em um modelo computacional que decompõe a imagem nas componentes de brilho e cromaticidade, leva em consideração as características dos *pixels*, as variações no ambiente e na cena e as variações da câmera, o que torna o método bastante simples e robusto.

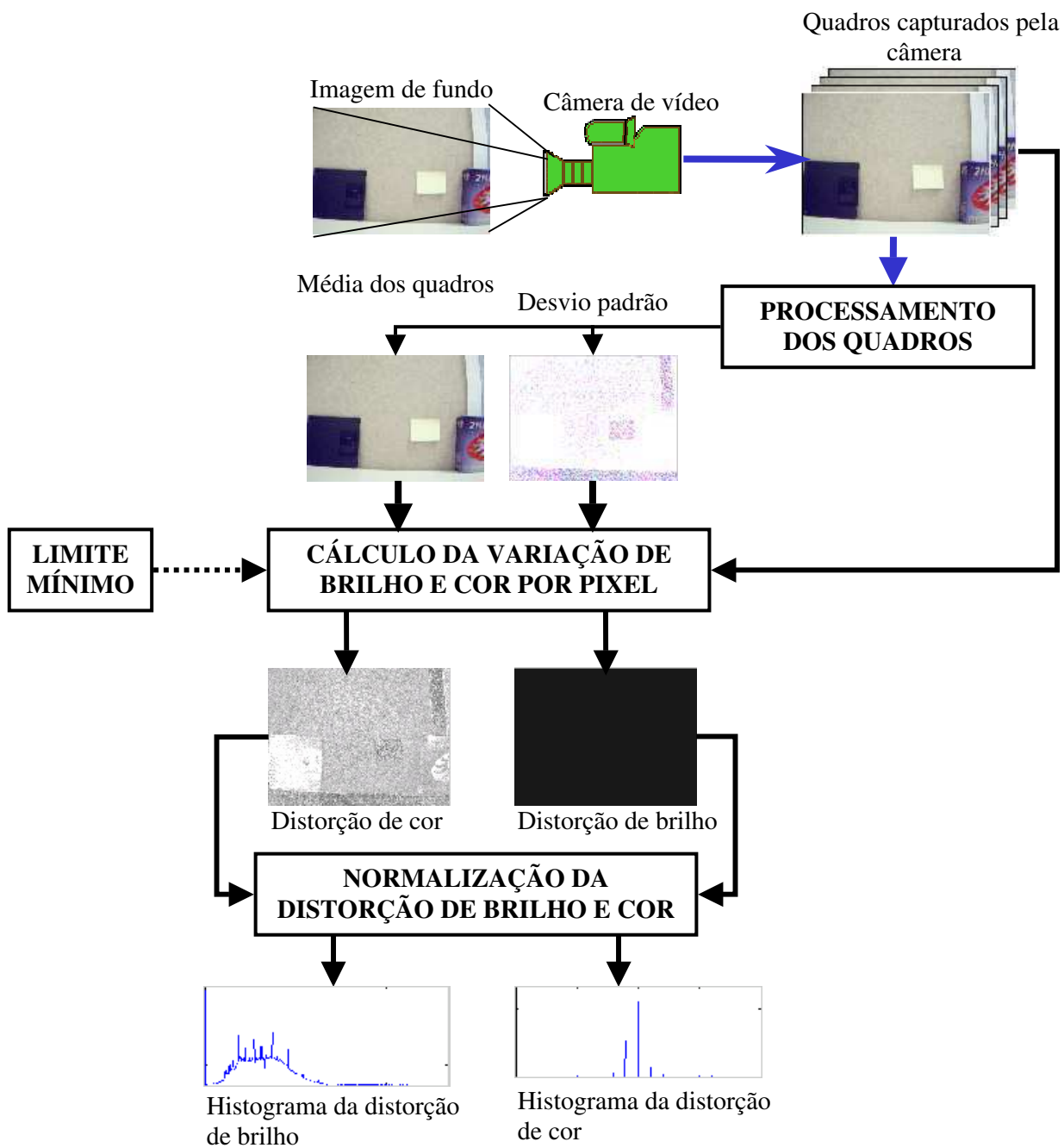


Figura 2.7.9: Algoritmo de extração de fundo. Etapa de modelagem de fundo.

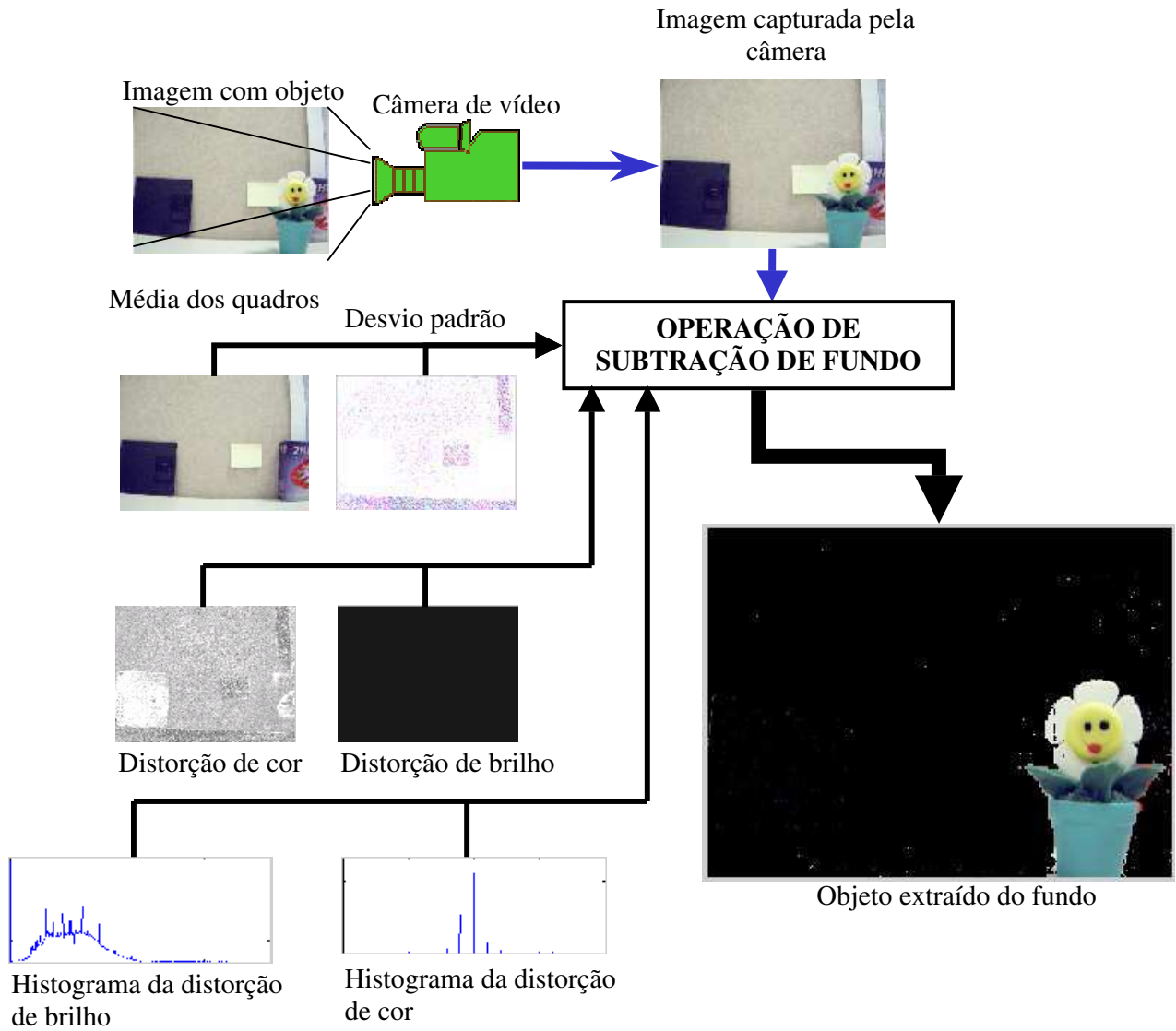


Figura 2.7.10: Algoritmo de extração de fundo. Etapa de extração de objeto.

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.

3

ALGORITMO DE HORPRASET E PLATAFORMA DSP

Neste capítulo apresenta-se de forma bem resumida o algoritmo de extração de imagem implementado na plataforma DSP da *Analog Device*, os fundamentos teóricos deste método e as características técnicas do *kit* de desenvolvimento utilizado.

3.1 INTRODUÇÃO

Foi escolhido para a implementação deste trabalho o algoritmo de Horpraset *et al.* [8], pois utiliza um fundo estático para a extração do objeto e o procedimento estatístico é extremamente simples de ser implementado, além de ser bastante robusto na detecção e eliminação de sombras, bem como a baixa sensibilidade às variações de iluminações globais e locais do ambiente.

3.2 MODELO COMPUTACIONAL

No algoritmo apresentado por Horpraset *et al.* [8], o modelo é baseado na diferença entre as cores medida pelas distorções de brilho (*brightness*) e de cromaticidade (*chromaticity*). A Figura 3.2.11 ilustra o modelo computacional de cor proposto neste algoritmo no espaço tridimensional RGB. Considerando o ponto (*pixel*) i na imagem, denota-se $E_i = [E_R(i), E_G(i), E_B(i)]$ como sendo o vetor que representa a cor esperada do ponto (imagem de referência). A linha $\overline{OE_i}$, que passa pela origem e pelo ponto E_i , é chamada de *linha de cor (expected chromaticity line)*. $I_i = [I_R(i), I_G(i), I_B(i)]$ é a cor RGB do ponto na imagem atual, da qual se quer subtrair o fundo. Neste modelo, basicamente se quer medir a distorção existente entre I_i e E_i . Isto é possível medindo a distorção de brilho e de cor, no valor RGB do *pixel*.

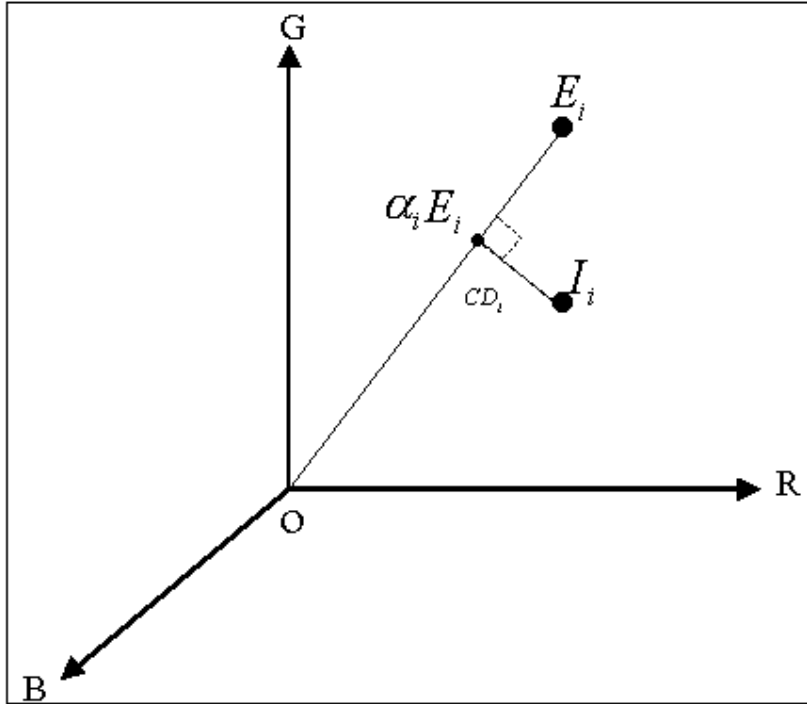


Figura 3.2.11: Modelo de cor no espaço tridimensional RGB proposto em [8]. E_i representa o valor de cor esperado do ponto i na imagem de referência; I_i representa a cor na imagem atual. A diferença entre I_i e E_i é decomposta em distorção de brilho (α_i) e distorção de cor (CD_i).

A distorção de brilho (α_i) é o valor escalar que projeta a cor atual I_i na linha de cor de modo que ela se aproxime o máximo possível do valor de referência E_i . α_i é o argumento que minimiza:

$$\phi(\alpha_i) = (I_i - \alpha_i E_i)^2 \quad (4.1)$$

Se α_i é menor do que 1, este é mais escuro. Se α_i é maior do que 1, este é mais claro. A distorção e cor CD_i são definidas como sendo a distância ortogonal entre o valor atual I_i e a linha de cor no modelo. Este é dado por:

$$CD_i = \|I_i - \alpha_i E_i\| \quad (4.2)$$

Para entender melhor este modelo, são importantes algumas considerações: a distorção de brilho é um valor escalar que posiciona o valor da cor atual ao longo da linha de cor. Uma característica importante desta linha é que ela mantém certo padrão de cor ao longo dela, mudando apenas a intensidade luminosa. Assim, é possível determinar se certo ponto na imagem tem um valor de cor parecido com outro, mas com brilho diferente. A distorção de brilho significa apenas uma mudança na iluminação e não uma mudança significativa na cor do *pixel*. Esta diferenciação permite identificar pontos da imagem que representam sombras, pois eles têm um brilho bem menor do que o original no modelo de referência, mas ainda mantêm uma cor parecida à original. A distorção de cor indica o grau de afastamento que se tem dos padrões de cores representados pela linha. Assim, por este indicador pode-se saber se a cor atual está muito longe ou não do padrão de cores representado pela linha, o que é útil para saber se aquele *pixel* faz parte de um objeto diferente que está presente na imagem atual.

3.3 MODELAGEM DO FUNDO

No processo de treinamento do algoritmo apresentado por Horprasert et al. [8], a imagem de referência e alguns atributos relacionados com ela são calculados ao longo de um período de tempo ou com base em um número definido de imagens de treino. Cada *pixel* é modelado como sendo um vetor de 4 componentes, $\langle E_i, S_i, a_i, b_i \rangle$, em que E_i é o valor esperado da cor, S_i é o desvio padrão, a_i é a distorção média de brilho e b_i é a distorção média de cor. Neste algoritmo, E_i é considerado como sendo a imagem média das imagens de treino e S_i como sendo o desvio médio padrão presente neste conjunto de imagens. Ou seja:

$$E_i = [\mu_R(i), \mu_G(i), \mu_B(i)] \quad (4.3)$$

Onde, $\mu_R(i)$, $\mu_G(i)$ e $\mu_B(i)$ são as médias aritméticas dos valores de R, G e B do *pixel* i tomados em N quadros.

$$S_i = [\sigma_R(i), \sigma_G(i), \sigma_B(i)] \quad (4.4)$$

Para o cálculo de a_i e b_i são consideradas as seguintes equações:

$$\alpha_i = \frac{\left(\frac{I_R(i)\mu_R(i)}{(\sigma_R(i))^2} + \frac{I_G(i)\mu_G(i)}{(\sigma_G(i))^2} + \frac{I_B(i)\mu_B(i)}{(\sigma_B(i))^2} \right)}{\left(\left(\frac{\mu_R(i)}{\sigma_R(i)} \right)^2 + \left(\frac{\mu_G(i)}{\sigma_G(i)} \right)^2 + \left(\frac{\mu_B(i)}{\sigma_B(i)} \right)^2 \right)} \quad (4.5)$$

$$CD_i = \sqrt{\left(\frac{I_R(i) - \alpha_i \mu_R(i)}{\sigma_R(i)} \right)^2 + \left(\frac{I_G(i) - \alpha_i \mu_G(i)}{\sigma_G(i)} \right)^2 + \left(\frac{I_B(i) - \alpha_i \mu_B(i)}{\sigma_B(i)} \right)^2} \quad (4.6)$$

$$a_i = RMS(\alpha_i - 1) = \sqrt{\frac{\sum_{i=0}^{N-1} (\alpha_i - 1)^2}{N}} \quad (4.7)$$

$$b_i = RMS(CD_i) = \sqrt{\frac{\sum_{i=0}^{N-1} (CD_i)^2}{N}} \quad (4.8)$$

O processo de treino e o cálculo das imagens de referência do fundo no algoritmo de Horprasert et al. [8], foram mostrados na Figura 2.7.9 e Figura 2.7.10. O processo começa com a captura de um número de imagens de treino, a partir das quais se calcula uma imagem média ($E_i = [\mu_R(i), \mu_G(i), \mu_B(i)]$) e outra representando o desvio padrão ($S_i = [\sigma_R(i), \sigma_G(i), \sigma_B(i)]$).

Para cada uma das imagens de referência calculam-se as imagens da distorção de brilho (α_i) e da distorção de cor em relação à imagem média (CD_i); para obter estas

imagens, utilizam-se as Equações (4.5) e (4.6) respectivamente. Para o cálculo destas imagens, são utilizadas as imagens originais, a imagem média, o desvio padrão e as Equações (4.5) e (4.6). Utiliza-se a Equação (4.7) e (4.8) para o cálculo da distorção de brilho e cor, sendo normalizados e gerados os histogramas destas distorções para os limiares de comparação, completando assim o processo de treinamento.

No final do processo de treino no algoritmo [8], são obtidas 4 imagens de referência: a imagem média, a imagem do desvio padrão médio, a imagem da distorção média de brilho e a imagem da distorção média de cor. Nestas imagens, estão modeladas as características do fundo, e são estas as utilizadas como referência no processo de subtração.

Nas Equações (4.5) e (4.6), é calculada a distorção de brilho e de cor. Pode-se ver que, no cálculo, estes valores são normalizados ou divididos pelo valor do desvio padrão [8]; assim, *pixels* com baixo desvio padrão podem ter valores de distorção de brilho e cor muito altos. Para solucionar este problema, é importante determinar um valor mínimo do desvio padrão que pode ter cada um dos pontos nas imagens. Não existe um padrão de atribuição para este valor no algoritmo [8], portanto este valor é atribuído por um método de tentativa e erro (testando diferentes valores positivos menores ou iguais a 1). Isso foi feito na implementação do algoritmo.

3.4 EXTRAÇÃO DE OBJETO

Nesta etapa de extração do objeto, conforme Figura 2.7.10, é calculada a diferença existente entre a imagem de referência e a imagem atual. No algoritmo apresentado por Horprasert et al.[8], essa diferença é decomposta em distorção de brilho e distorção de cor. Aplicando um limiar apropriado na comparação da distorção de brilho e da distorção de cor em um *pixel*, pode-se obter uma máscara indicando qual é o tipo do *pixel* que está sendo comparado. De acordo com o modelo de decomposição de cor que é utilizado na fase de subtração, cada *pixel* pode ser classificado em uma das 4 categorias seguintes:

- **Fundo original (B)**, se a distorção de brilho e a distorção de cor são similares aos valores correspondentes na imagem de referência.

- **Fundo com baixa iluminação ou sombra (S)**, se a distorção de cor é similar, mas a distorção de brilho é bem menor que na imagem de referência.
- **Fundo iluminado (H)**, se existe uma distorção de cor similar, mas uma distorção de brilho maior que na imagem de referência.
- **Objeto (F)**, se a distorção de cor é diferente dos valores esperados no treinamento, caso em que o *pixel* é considerado como sendo parte do objeto.

A Figura 2.7.10 ilustra a etapa de subtração deste algoritmo. A imagem que contém o objeto é comparada com a imagem média e com o desvio padrão para obter imagens que representam a distorção de brilho e a distorção de cor, para o que se aplicam as Equações (4.5) e (4.6) respectivamente. As imagens são comparadas com as imagens de referência aplicando valores apropriados do limiar (*threshold*) de comparação. Pontos pertencentes a sombras ou pontos com maior iluminação foram considerados como sendo parte do fundo. A comparação da imagem atual com a imagem de referência é feita basicamente com base nas diferenças existentes entre os valores das distorções de brilho e das distorções de cor.

3.5 HARDWARE UTILIZADO

Para a implementação do algoritmo de Horprasert et al.[8] foi utilizada a plataforma de desenvolvimento ADSP-BF 5xx EZ-KIT Lite da Analog Device [1], que utiliza como processador de sinais o Blackfin BF 533.

Essa plataforma suporta entradas e saídas de vídeo analógico. O encoder de vídeo utilizado nesta placa é o ADV7171 da Analog Device. Este chip tem três entradas de vídeo multiplexadas e utiliza o espaço YCbCr para amostrar o sinal digital. Nesta mesma placa de desenvolvimento, tem-se um decoder de vídeo denominado de ADV7183, fabricado também pela Analog Device. Esse dispositivo multiplexa até três saídas de vídeo. Tanto o decodificador como o codificador de vídeo opera em 27 MHz e se conectam ao processador da placa por intermédio da PPI (*Parallel Peripheral Interface*).

A plataforma ADSP-BF533 EZ-KIT Lite foi projetada para utilizar todos os recursos do processador Blackfin BF533. Sua tensão de trabalho é de 5V ou 3.3V, mas pode variar até 1.4V, dependendo da frequência de trabalho que foi configurada. Nessa plataforma têm-se vários dispositivos que compõem e auxiliam no processamento desta placa, como é mostrado na Figura 3.5.12. Este trabalho detém-se nos dispositivos que integram as partes de aquisição, processamento e amostragem do sinal de vídeo, objeto desta dissertação.

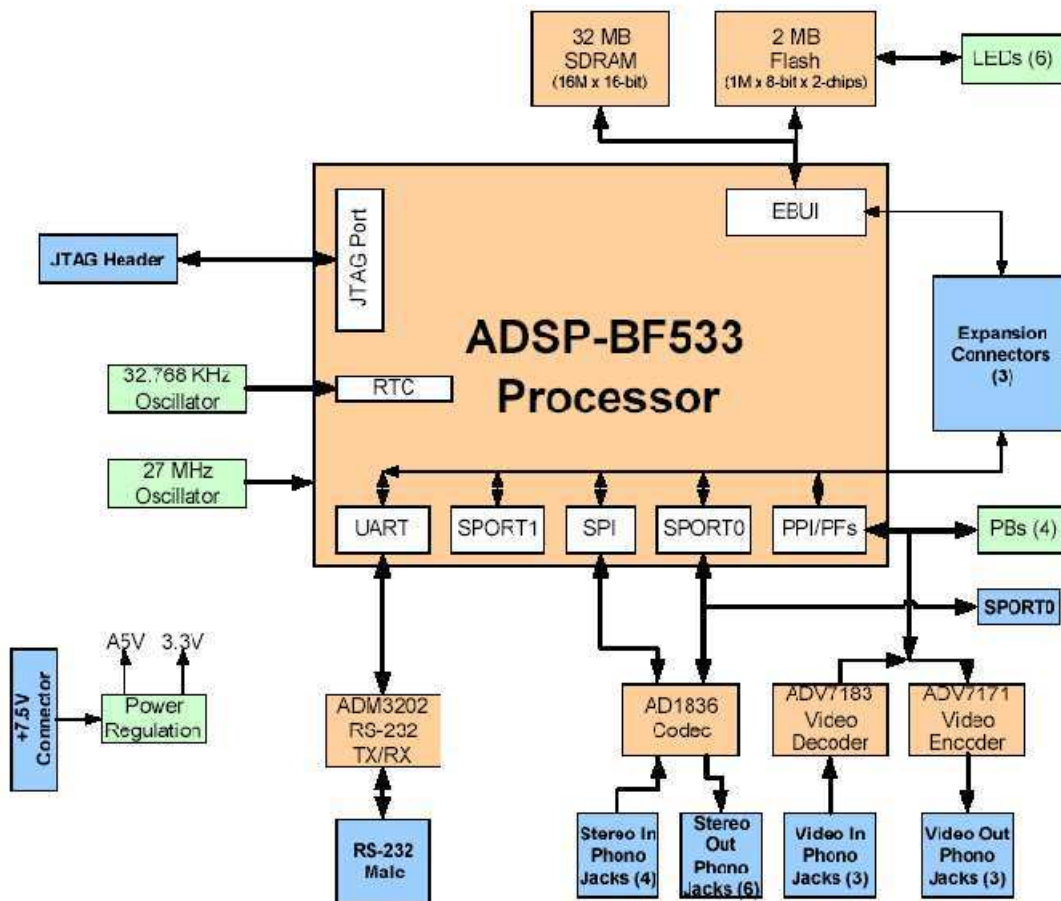


Figura 3.5.12: Arquitetura do ADSP-BF533 EZ-KIT Lite.

Blackfin é uma família de DSP's que foi desenvolvida entre as empresas Intel Corporation e Analog Device [1]. Esses processadores são utilizados principalmente em aplicações voltadas para as áreas de multimídia e telecomunicações. São destinados a

aplicações que requerem grande quantidade de manipulação de dados, baixos custos, baixo consumo de energia e, principalmente, aplicações que requerem uma grande capacidade computacional para processar sinais de áudio e vídeo em tempo real.

O processador Blackfin é um DSP de 16 bits que possui um conjunto de 8 registradores de 32 bits, em que cada um pode operar como dois registradores independentes de 16 bits. Esses processadores têm uma alta performance, com *clock* de 756 MHz, atingindo 3 GMAC's. Esse desempenho é possível devido ao seu *core*³ ser duplicado. Uma outra característica de fundamental importância na implementação deste trabalho foi o controlador da PPI (*Parallel Peripheral Interface*) apresentar suporte para o formato de vídeo ITU-R⁴ 656, adaptando assim o algoritmo inicialmente proposto, visando à implementação nessa plataforma.

3.5.1 Arquitetura Blackfin

Os DSP's Blackfin combinam uma arquitetura interna (Figura 3.5.13) de duplo MAC's⁵ (*Multiply-Add-Accumulate*) com a facilidade de programação de um microcontrolador tipo RISC (*Reduced Instruction Set Computer*), processando suas instruções em modo SIMD⁶ (*Single Instruction Multiple Data*), aumentando significativamente sua performance.

O *core* do processador (Figura 3.5.13) possui duas unidades importantes:

Unidade Aritmética de Dados: nesta unidade, são executadas todas as operações lógicas e aritméticas, bem como as operações com deslocamento de *bits* (registradores de deslocamento).

Unidade Aritmética de Dados: Esta unidade é responsável em gerar os endereços para o duplo acesso na memória do DSP.

³ Centro ou Núcleo. O "core" é a pastilha de silício que contém todos os componentes do processador.

⁴ Código da norma editada pela divisão de radiocomunicação do *International Telecommunication Union*.

⁵ São sistemas complexos, capazes de executar operações de multiplicação, soma, deslocamento e acumular o resultado em *pipeline*, com a execução em média de uma operação por ciclo de relógio.

⁶ Trata-se de uma arquitetura de computadores que executa uma operação sobre conjuntos múltiplos de dados.

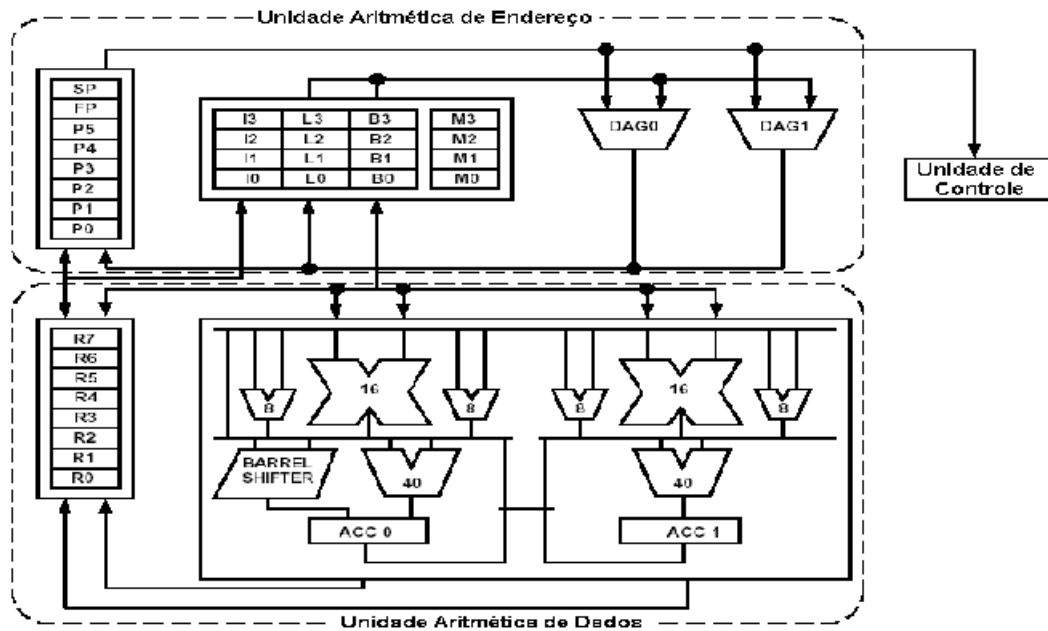


Figura 3.5.13: Core do DSP Blackfin.

3.5.2 Ambiente de desenvolvimento

Inicialmente, os algoritmos de extração de imagens foram desenvolvidos e testados no programa *Matlab* [15], de forma a validar as técnicas utilizadas. Os resultados obtidos foram comparados com os conseguidos nos trabalhos de Printes [17] e Parentes [16].

Uma vez validado o modelo de simulação, a mesma estrutura de algoritmos foi desenvolvida utilizando-se o ambiente integrado de desenvolvimento Visual DSP 4.0 (Figura 3.5.14), que possui compilador C/C++ e depurador para a plataforma de desenvolvimento ADSP-BF 5xx EZ-KIT Lite, fabricada pela *Analog Devices* [1] e a Intel Corporation. Essa plataforma de desenvolvimento se conecta a uma porta USB de um computador pessoal (PC) de modo a interagir entre si.

Algoritmo para Extração de Imagens de Fundos não-Homogêneos Usando o Espaço de Cores YCbCr

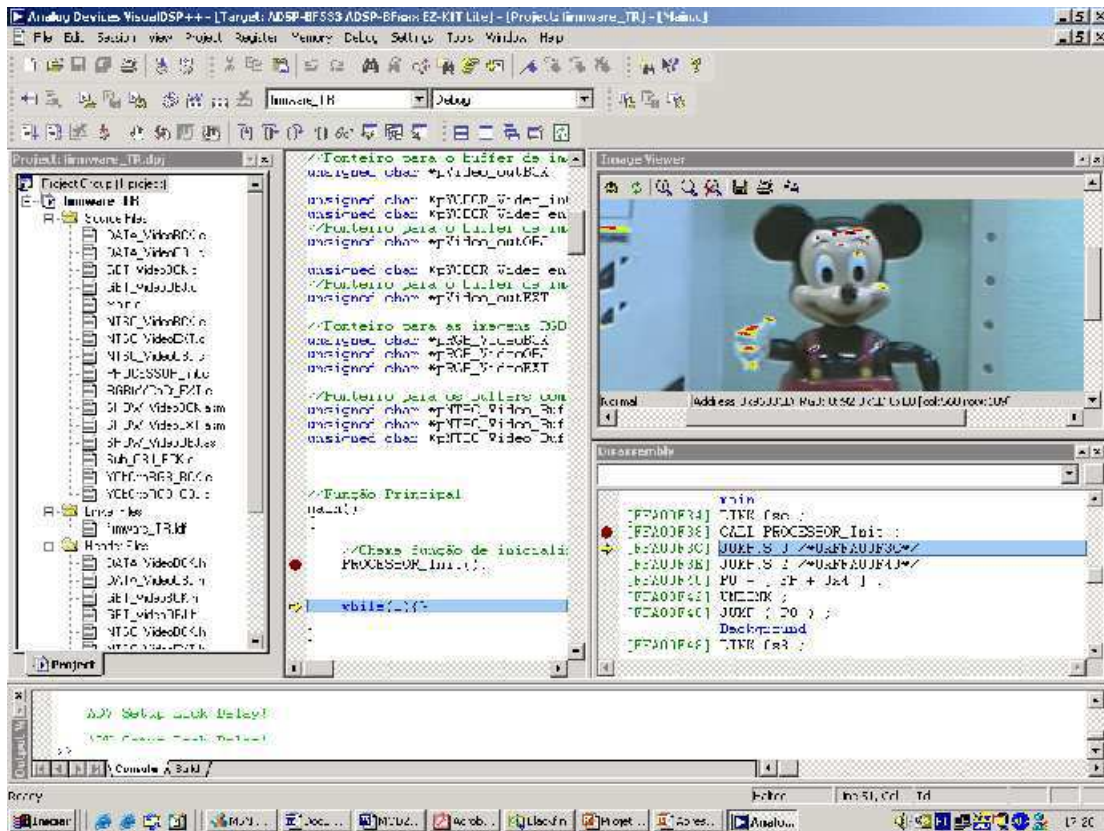


Figura 3.5.14: Ambiente de desenvolvimento Visual DSP++ 4.0.

Os programas desenvolvidos no ambiente Visual DSP 4.0 podem ser executados diretamente nesta placa. Ela possui vários recursos que ajudam no desenvolvimento do projeto, tais como: montador *assembler*, simuladores e várias bibliotecas de alto desempenho para a arquitetura Blackfin.

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.

4

EXTRAÇÃO DE IMAGEM UTILIZANDO O ESPAÇO DE CORES YCbCr

Neste capítulo, define-se o espaço de cor YCbCr que é usado no método proposto. São estabelecidas as relações entre os vetores, bem como as métricas e os parâmetros estatísticos utilizados.

4.1 INTRODUÇÃO

A maioria dos trabalhos de extração de imagens [8,17,18] utiliza no seu bojo espaço de cores RGB, devido à sua facilidade computacional. Propõe-se um novo método para extração de objetos em fundos não-homogêneos e desconhecidos, utilizando-se diretamente o espaço de cores YCbCr. Esta proposta visa atender *hardware* de desenvolvimento de processamento de imagens que utiliza *encoder* e *decoder* de vídeo, que convertem sinais de vídeo composto em sinais no formato YCbCr 4:2:2 ou 4:4:4, evitando, assim, conversões do sistema RGB para YCbCr e vice-versa.

4.2 REPRESENTAÇÃO DO ESPAÇO DE CORES YCbCr

A percepção das cores depende das quantidades de cores vermelha, verde e azul distinguidas pela visão. Esses valores são independentes, podendo variar de várias maneiras: uma cor pode ser luminosa ou não. Quando ela não tem iluminação, diz-se que seu valor é 1, e, quando tem, seu valor é 0. O espaço de cores RGB mostrado na Figura 4.2.15 pode ser representado por um cubo unitário. Neste espaço, estão todas as cores que podem ser representadas por RGB, cuja origem é 0,0,0, em que essa coordenada representa a cor branca; a cor preta tem como coordenada 1,1,1. Outras cores têm como coordenadas: cores primárias vermelha (1,0,0), verde (0,1,0) e azul (0,0,1). As cores complementares: ciano (0,1,1), magenta (1,0,1) e amarelo (1,1,0) [5].

Atualmente, o formato digital mais utilizado na representação de imagens é YCbCr. A sua conversão para o padrão RGB é determinada pela Recomendação ITU-R BT.601, adotada em 1990. A luminância e a crominância no sinal digital do espaço YCbCr são modeladas e projetadas para facilitar o intercâmbio com outros espaços de cores usados no padrão analógico internacional [28].

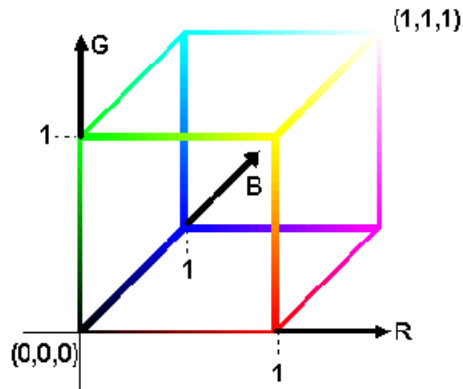


Figura 4.2.15: Cubo unitário do espaço de cores RGB.

Na representação do espaço de YCbCr mostrado na Figura 4.2.16, a luminância é denotada pelo Y, que pode ser descrita como uma combinação unitária do vermelho, verde e azul (conforme foi mostrado no capítulo 2). Sua faixa de atuação vai do 0 (preto) ao 1 (branco); enquanto a crominância representada por Cb e Cr varia de $-1/2$ a $1/2$. Vale ressaltar que os três valores Y, Cb e Cr são independentes entre si.

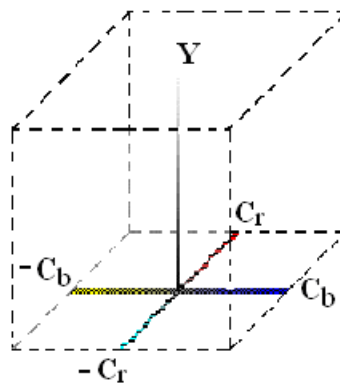


Figura 4.2.16: Cubo unitário do espaço de cores YCbCr.

Assim como no espaço de cores RGB, o espaço Y'CbCr pode ser representado como um cubo unitário. Mas aqui o eixo da luminância não se estende ao longo da diagonal principal, mas a cor preta está na origem $[Y',Cb,Cr] = [0,0,0]$, que é o centro de uma das

faces do cubo. A cor branca está no centro da face oposta (1,0,0) do cubo. O eixo de Cb varia do amarelo de -1/2 a 0. A outra extremidade da face contém a origem do azul, que vai de 0 a 1/2. Na extremidade oposta, o eixo de Cr varia de um ciano que vai de -1/2 a 0. A outra extremidade dessa face contém a origem do vermelho, que varia de 0 a 1/2.

A norma BT.601 define que a luminância Y tenha uma variação nominal de 16-235 (que vai da cor preta à branca), na qual zero (preto) corresponde a 128; Na crominância, são definidos Cb e Cr para terem uma variação nominal de 16 - 240. A conversão do espaço de cores RGB para YCbCr é proposta nesta mesma norma, que está como anexo na dissertação.

4.3 ALGORITMO PROPOSTO

O modelo computacional utilizado baseia-se em uma característica da visão humana conhecida como constância de cor [3]. A visão humana percebe uma cor constante em um objeto, mesmo se submetido a variações de luminosidade no tempo e espaço. Esta constatação possibilita a detecção de sombras em um fundo conhecido, desde que se consiga detectar variações de cor associadas às variações de luminosidade [8].

Propõe-se um modelo tridimensional com as componentes Y, Cb e Cr, no qual são representados todos os pontos relativos a uma determinada cena, neste caso, com seus valores discretizados. Este modelamento possibilita a representação de cada *pixel* como uma composição de dois termos separados, brilho e cromaticidade como ilustrado pela Figura 4.3.17. Mostra-se uma representação gráfica do modelo proposto com seus três eixos ortogonais Y,Cb e Cr, e um ponto E_i qualquer (Figura 4.3.17a).

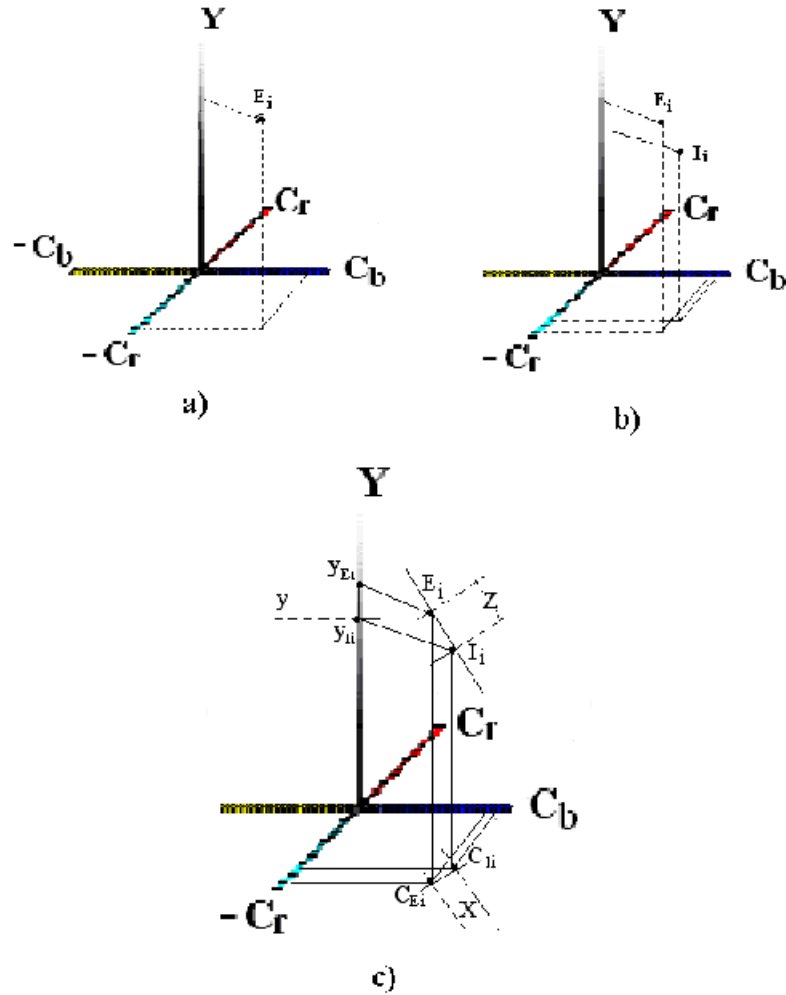


Figura 4.3.17: Espaço de cores YCbCr: a) E_i a média de dos *pixels* da imagem de fundo; b) I_i *pixel* da imagem objeto; c) Relação métrica entre os *pixels* E_i e I_i .

O vetor E_i representa a média dos *pixels* na posição i dos N quadros da imagem de referência ou imagem de fundo (Figura 4.3.17a). O vetor I_i representa o *pixel* i do quadro da imagem com objeto que se quer extrair.

Considere um *pixel* E_i qualquer na imagem. Suponha,

$$E_i = [y_{Ei}, C_{Ei}], \quad (5-1)$$

o valor esperado para o *pixel* naquele ponto da imagem de referência. Também é mostrada a representação do *pixel* I_i , como:

$$I_i = [y_{ii}, C_{ii}] \quad (5-2)$$

que representa o mesmo *pixel* na imagem corrente, o qual deseja extrair da imagem de referência.

4.4 AQUISIÇÃO DAS IMAGENS

A primeira fase deste método consiste na aquisição de uma imagem de referência, que se chama imagem de fundo, cujo *pixel* médio denomina-se de E_i , que fará parte da imagem média. Esta fase é conhecida como fase de aprendizagem.

Na segunda fase deste método, é obtida uma nova imagem que é comparada com a imagem de referência, obtida na fase de aprendizagem, para que seja feita a extração da imagem de fundo. Isto é feito calculando-se as distorções de brilho, a distorção de cor e a distância entre os *pixels* E_i e I_i .

Na Figura 4.3.17c, a linha Z representa a distância entre o vetor E_i e I_i . A distorção de cor é calculada medindo-se a distância da projeção dos vetores E_i e I_i no plano $C_b C_r$ e chamar-se-á X. A distância entre os pontos da projeção dos vetores E_i e I_i no eixo da luminância Y é denominada de distorção de luminância, que se designa pela letra y.

4.5 MODELAGEM DO FUNDO

O algoritmo necessita de uma fase de aprendizagem que consiste na aquisição de alguns quadros da imagem de fundo, a fim de ser obtida a média dos *pixels* que servirá como imagem de referência. O Parâmetro E_i é a media dos N quadros obtidos para cada um

dos *pixels*, em cada componente de luminância Y e crominância CbCr. Portanto, sendo P_i o *pixel* da imagem de um quadro, então:

$$E_i = [W_i(Y), W_i(C_b), W_i(C_r)] = [W_i(\eta)] \quad (5-3)$$

no qual $\eta = Y, C_b, C_r$

$$E_i = [W_i(\eta)] = \left[\frac{1}{N} \sum_{i=0}^{N-1} P_i(\eta) \right] \quad (5-4)$$

em que $P_i(\eta)$ representa o valor do *pixel* i , $W_i(\eta)$ o valor médio do *pixel* i e N o número de quadros amostrado de uma imagem.

4.6 DEFINIÇÃO DA MÉTRICA UTILIZADA

Sejam E_i e I_i vetores no espaço \mathfrak{R}^3 , definem-se duas métricas neste espaço:

4.6.1 Métrica do máximo [30].

A norma é a forma de medir a distância entre vetores. A métrica do máximo é definida como:

$$d(E_i, I_i) = \|E_i - I_i\|_M = \text{Max} \{ |E_i - I_i| \} \quad (5-5)$$

no qual, d é a distância entre os *pixels*.

Sabe-se que esta forma de medir é equivalente a métrica Euclidiana $\|E_i - I_i\|$.

4.6.2 Métrica da Soma [30].

A métrica da soma é definida como:

$$d(E_i, I_i) = \|E_i - I_i\|_s = \sum |E_i - I_i| \quad (5-6)$$

Sabe-se que estas formas de medir (5-5) e (5-6) são equivalentes à métrica Euclidiana $\|E_i - I_i\|$. Portanto,

$$\|E_i - I_i\|_M \leq \|E_i - I_i\| \leq \|E_i - I_i\|_s \quad (5-7)$$

Para o cálculo da distância entre os pontos, adota-se a métrica do máximo neste trabalho, devido ao esforço computacional ser menor em relação à métrica Euclidiana e à métrica da soma.

4.7 EXTRAÇÃO DE OBJETO

A etapa de extração de objeto é feita diretamente no espaço YCbCr 4:4:4. Cada *pixel* da imagem do primeiro plano é comparado com o seu equivalente na imagem de fundo. É calculada a distância entre os *pixels* E_i e I_i através da métrica do máximo que servirá como parâmetro de comparação. Assim:

$$Z = \| E_i - I_i \| = \text{Max} \{ |E_i - I_i| \} \quad (5-8)$$

em que Z é a métrica do máximo no \mathfrak{R}^3 .

O mesmo processo é aplicado para calcular a distorção de crominância entre os *pixels* E_i e I_i , que equivale a medir a distância entre os pontos C_{E_i} e C_{I_i} no plano CbCr. Portanto:

$$X = \| C_{E_i} - C_{I_i} \| = \text{Max} \{ |C_{E_i} - C_{I_i}| \} \quad (5-9)$$

em que X é a métrica do máximo no \mathfrak{R}^2 .

Analogamente,

$$y = \| y_{E_i} - y_{I_i} \| = \text{Max} \{ |y_{E_i} - y_{I_i}| \} \quad (5-10)$$

em que y é a métrica do máximo no eixo Y.

Seja I_i o mesmo *pixel* no primeiro plano da imagem de frente, esse *pixel* pode assumir dois valores, a saber:

- Se $E_i = I_i$: pode-se afirmar que a distância entre esses dois vetores é igual a zero.
- Se $E_i \neq I_i$: pode-se afirmar que a distância entre esses dois vetores é diferente de zero.

4.7.1 Fundo original.

Quando: $E_i = I_i$, pode-se afirmar que:

- a) $y_{E_i} = y_{I_i}$
- b) $C_{E_i} = C_{I_i}$.

Pode-se ver na Figura 4.7.18 que a distância entre esses dois vetores é igual a zero, ou seja, $Z = X = y = 0$. Para esse caso, classifica-se esse *pixel* como sendo fundo original.

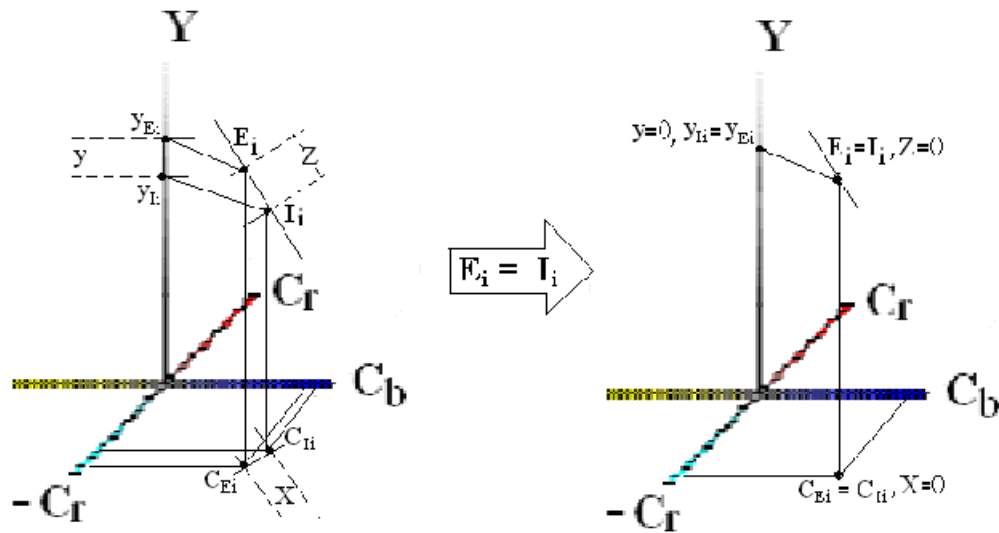


Figura 4.7.18: Fundo original, quando $Z = X = y = 0$

Quando $E_i \neq I_i$, pode-se afirmar que $Z \neq 0$, com as seguintes situações:

4.7.2 Objeto.

Caso 1:

Quando a distorção de luminância é igual a zero e distorção de crominância diferente de zero, observa-se na Figura 4.7.19 que, para este caso, a distância entre y_{E_i} e y_{I_i} é igual a zero e a distância entre C_{E_i} e C_{I_i} é diferente de zero. Portanto, pode-se afirmar que a luminância é a mesma para o mesmo *pixel*, ou seja $y_{E_i} = y_{I_i}$ e $C_{E_i} \neq C_{I_i}$, assim, $y = 0$ e $X \neq 0$. Para esse caso, classifica-se o *pixel* como objeto. Vale ressaltar que o limite para essa distorção é a média do desvio padrão entre os *pixels*.

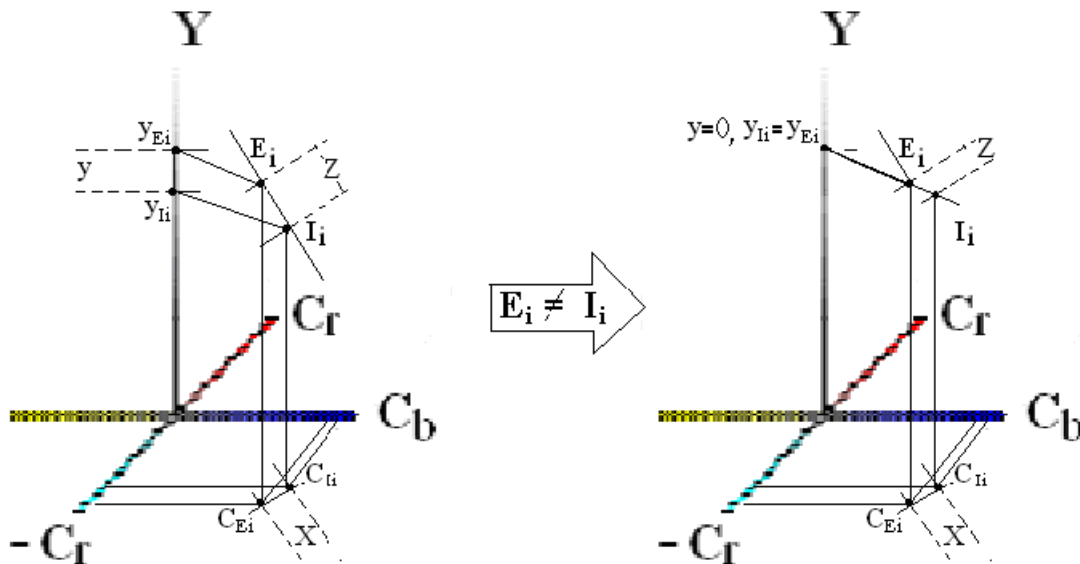


Figura 4.7.19: *Pixel* definido como objeto, quando $Z \neq 0$, $X \neq 0$ e $y = 0$

Caso 2:

Quando a distorção de luminância for diferente de zero e distorção de crominância diferente de zero. Neste caso, observa-se que $y_{E_i} \neq y_{I_i}$ e $C_{E_i} \neq C_{I_i}$, ou seja, $y \neq 0$ e $X \neq 0$. Para esse caso, classifica-se o *pixel* também como objeto.

4.7.3 Sombra ou iluminação.

Quando a distorção de luminância for diferente de zero e distorção de crominância igual a zero. Observa-se na Figura 4.7.20 que, nesse caso, $Z = y$, no qual $Z \neq 0$. A crominância entre esses dois *pixels* é igual. Assim, $X = 0$, pois $C_{Ei} = C_{Ii}$. Para esse caso, classifica-se o *pixel* como sombra ou iluminação.

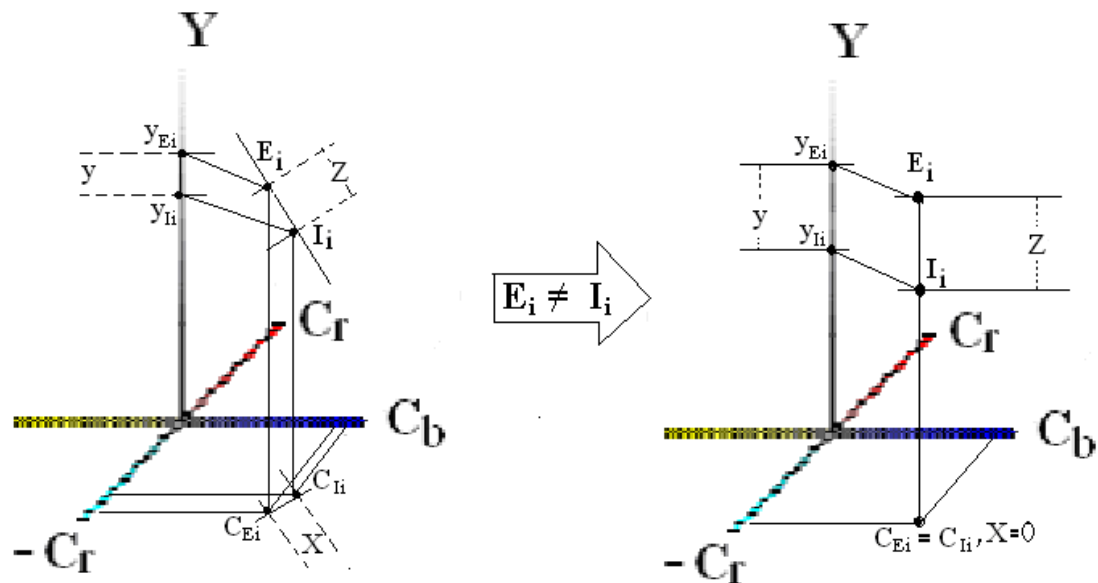


Figura 4.7.20: *Pixel* definido como sombra ou iluminação, quando $Z \neq 0$, $X=0$ e $y = Z$

Tomando y_{Ei} como referência, ou seja, transladando y_{Ei} para zero, pode-se classificar a polaridade de y_{Ii} : se y_{Ii} for negativo, o *pixel* é classificado como sombra; se y_{Ii} for positivo, o *pixel* é classificado como *pixel* iluminado.

4.7.4 Outros casos.

Distorção de luminância igual a zero e distorção de cromaticidade igual a zero. Esse caso é impossível de acontecer, pois inicialmente considerou-se $Z \neq 0$. Portanto, para $y = 0$ e $X = 0$, necessariamente $Z = 0$. Esse caso foi analisado anteriormente.

4.8 ABORDAGEM ESTATÍSTICA

Quando se analisa uma imagem estática capturada por um sensor CCD, raramente observa-se o mesmo valor para um mesmo *pixel* em determinado período de tempo. Este fato se deve principalmente a ruídos da câmera e flutuação na iluminação causada por variações das fontes de luz. Portanto, estas variações devem ser consideradas e mensuradas. Este é um dos principais motivos da necessidade de uma abordagem estatística para a extração de um objeto.

Não é suficiente a simples aquisição e análise de um único quadro da imagem de referência. São necessárias a aquisição e a análise de uma amostra estatisticamente significativa de quadros da imagem de referência conhecida, permitindo, assim, a determinação de parâmetros de normalização relativos às variações do sistema como um todo.

4.8.1 Desbalanceamento das cores.

Câmeras CCD's apresentam diferentes sensibilidades para cores diferentes, devido principalmente a diferenças na sensibilidade dos sensores RGB. Verificam-se respostas diferentes para a mesma variação de intensidade de cor de R, G e B. Dessa forma, torna-se necessário fazer o balanceamento ou normalização dos valores capturados de modo à equalizar as respostas [17]. Essas características são refletidas no espaço de cores YCbCr. No algoritmo proposto, a ponderação é realizada tomando-se o desvio padrão (S_i) de cada

um dos *pixels* da imagem de referência, calculado em um número N de quadros. O desvio padrão (S_i) é dado por:

$$S_i = [\sigma_Y(i), \sigma_{Cb}(i), \sigma_{Cr}(i)] \quad (5-11)$$

nos quais $\sigma_Y(i), \sigma_{Cb}(i), \sigma_{Cr}(i)$ são os desvios padrão de cada *pixel* tomados em N quadros da imagem de fundo.

Para melhorar a classificação dos *pixels*, foi adotada uma outra medida de dispersão, chamada de amplitude [31], essa medida necessita classificar o maior e menor *pixel* nos N quadros comparados.

$$a_i = \max[W_i(\eta)] - \min[W_i(\eta)] \quad (5-12)$$

$$a_i = \max[W_i(y), W_i(C_b), W_i(C_r)] - \min[W_i(y), W_i(C_b), W_i(C_r)] \quad (5-13)$$

em que:

a_i é a amplitude dos *pixels* comparados.

$\max[W_i(\eta)]$ é o maior *pixel* observado nos N quadros da imagem de referência.

$\min[W_i(\eta)]$ é o menor *pixel* observado nos N quadros da imagem de referência.

Portanto, para classificar o *pixel* como fundo ou objeto, deve-se levar em consideração o desvio padrão S_i e a amplitude a_i de cada *pixel*, pois a classificação do *pixel* depende desses intervalos de dispersão.

O algoritmo proposto nesta dissertação foi implementado no MATLAB para validação da idéia, levando-se em conta as restrições relativas a uma futura implementação em *hardware*. Os resultados obtidos são mostrados no próximo capítulo.

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.

5

RESULTADOS OBTIDOS

Neste capítulo são mostrados os resultados obtidos do sistema de extração de objetos implementado em DSP com fundo estático, em ambiente real. São apresentados os blocos lógicos que compõem o sistema, bem como análise dos resultados obtidos. Apresenta-se também o resultado da simulação do algoritmo proposto em YCbCr.

5.1 INTRODUÇÃO

Inicialmente, o algoritmo proposto por Horprasert et al. [8] foi testado no programa *Matlab* 7.0 [15] de forma a validar a técnica, visando, posteriormente à sua implementação na plataforma de desenvolvimento ADSP-BF 533 EZ-KIT Lite, utilizando o ambiente de programação Visual DSP 4.0.

5.2 SIMULAÇÃO DO ALGORITMO NO MATLAB

A codificação do algoritmo [18] no MATLAB seguiu a seqüência mostrada na Figura 5.1.21. Todos os parâmetros estatísticos foram calculados e arredondados para duas casas decimais; essa limitação foi adotada já visando à implementação na plataforma DSP, haja vista que a plataforma utilizada não contém processador de ponto flutuante e a quantidade de memória RAM ser limitada a 16 MB para palavra de dado de 16 *bits*.

Os testes foram realizados em seqüências de imagens de vídeo com resolução de 320 x 240 *pixels*, capturadas por uma câmera de vídeo Panasonic PV-610, conectada à entrada de vídeo da plataforma de desenvolvimento ADSP-BF 533 EZ-KIT Lite, que, por sua vez, é conectada via interface USB em um computador PC com processador AMD XP 2400+ e memória RAM 256 Mb de RAM.

A câmera captura 30 quadros por segundo. Cada quadro foi capturado via placa DSP e armazenado como arquivo de imagem BMP (bitmap ou mapeamento de *bits*). Esse formato de imagens é mais "puro". Ele não usa nenhum tipo de compressão e, por isso, a qualidade de imagem é a melhor possível. Essas imagens são armazenadas no computador para cálculo da média dos quadros, tarefa realizada pelo programa implementado no MATLAB.

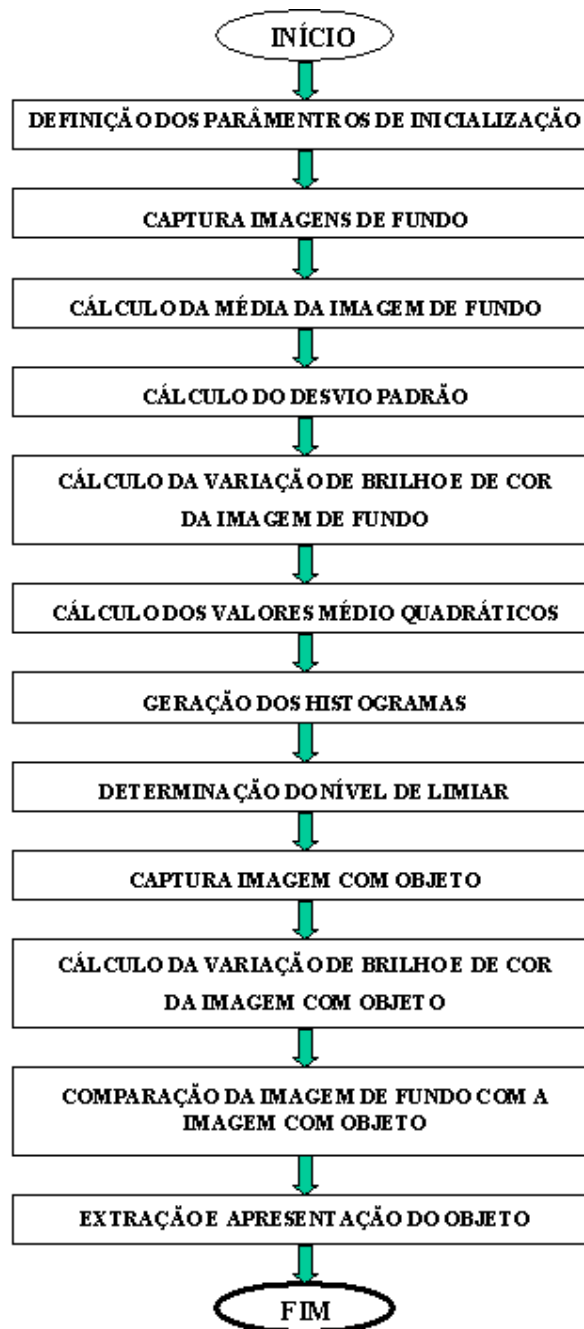


Figura 5.1.21: Diagrama do algoritmo implementado no MATLAB.

5.2.1 RESULTADOS OBTIDOS NA SIMULAÇÃO

São apresentados a seguir três resultados obtidos pela simulação da extração do objeto feita com o MATLAB 7.0. As imagens foram submetidas ao mesmo processo de extração de objeto, conforme a Figura 5.1.21.

A imagem da Figura 5.2.22 representa um caso no qual o fundo é uma parede pintada com tinta verde; neste caso, o algoritmo conseguiu tirar quase que todo o fundo. Porém, muito desses pontos estão muito próximos da imagem objeto, como a mão do boneco aqui representado, de modo que o algoritmo confunde como fundo, extraíndo uma boa parte da luva do boneco. Pode-se observar que outro ponto do objeto foi também confundido com o fundo, de modo que foi também extraído da imagem final.

A imagem da Figura 5.2.23 representa uma situação mais complexa, em que o fundo contém várias cores, o objeto inserido não apresenta sombra. Observa-se que o algoritmo retirou quase que completamente o fundo, mas também confundiu uma parte do rosto do objeto com o fundo.

A imagem da Figura 5.2.24 representa uma situação muito mais complexa, em que o fundo contém várias cores diferentes, e o objeto inserido apresenta sombra em várias direções. Assim como nos casos anteriores, observa-se que o algoritmo retirou quase que completamente o fundo, mas confundiu várias partes do objeto como fundo e não retirou as sombras por completo.

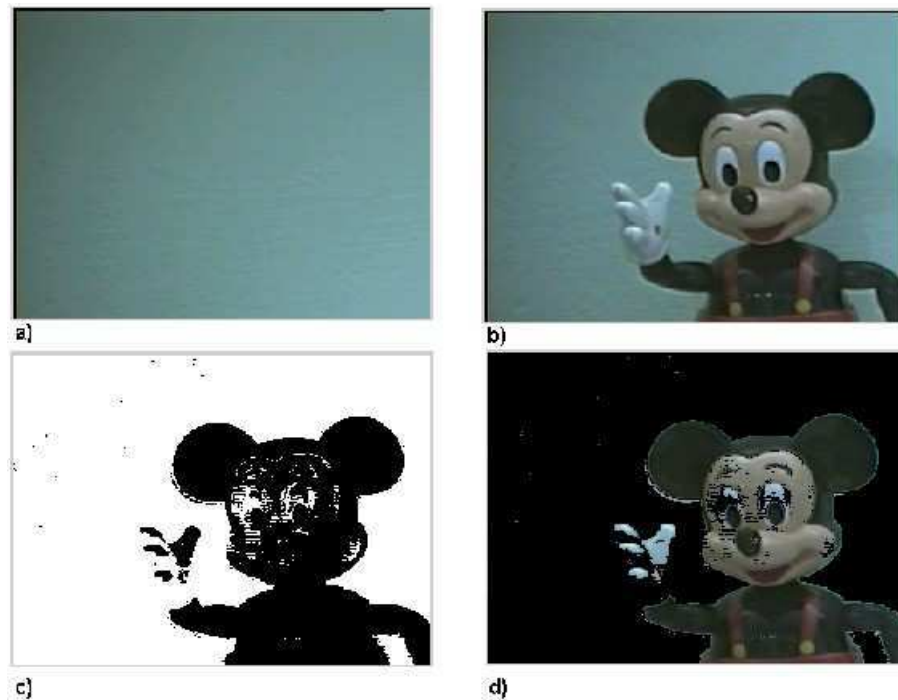


Figura 5.2.22: Resultado da simulação com MATLAB 7: a) Imagem média de fundo de baixa complexidade; b) Imagem objeto; c) Imagem binária para a extração do objeto; d) Imagem do objeto extraído.

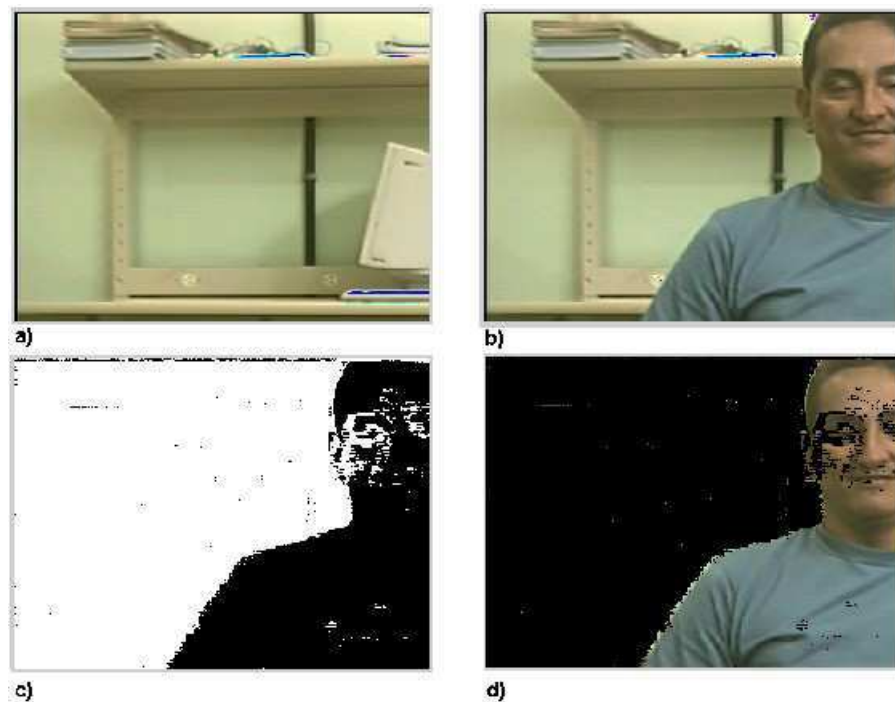


Figura 5.2.23: Resultado da simulação com MATLAB 7.0: a) Imagem média de fundo com vários objetos; b) Imagem objeto; c) Imagem binária para a extração do objeto; d) Imagem do objeto extraído.

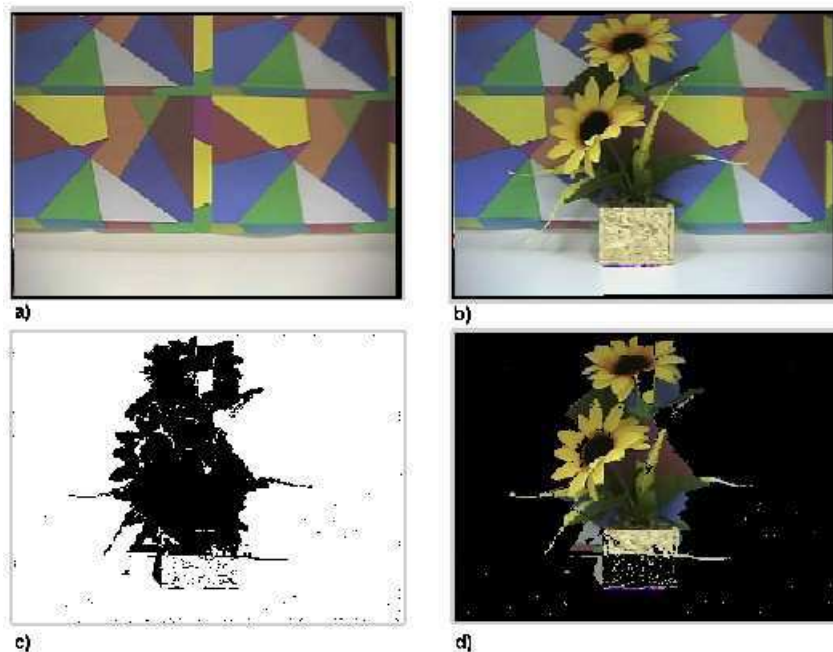


Figura 5.2.24: Resultado da simulação com MATLAB 7.0: a) Imagem média de fundo complexo; b) Imagem objeto; c) Imagem binária para a extração do objeto; d) Imagem do objeto extraído. Diagrama do algoritmo implementado no MATLAB.

5.3 SISTEMA DE EXTRAÇÃO DE OBJETOS IMPLEMENTADO EM DSP

Uma vez validado o modelo de simulação, a mesma estrutura de algoritmo foi desenvolvida e implementada utilizando-se o ambiente integrado de desenvolvimento Visual DSP 4.0, que possui compilador C, C++ e depurador para o processador ADSP-BF533 Blackfin da *Analog Device*.

No ambiente de programação Visual DSP 4.0 foram inseridos os códigos em C e C++ para configurar e programar a plataforma. O programa executado no *kit* de desenvolvimento captura, processa e apresenta a imagem na saída de vídeo, que, por sua vez, é conectada na televisão para visualizar a imagem extraída. A extração da imagem obedece ao fluxo simplificado da Figura 5.1.21.

A Figura 5.3.25 representa a imagem média do fundo capturado pela câmera de vídeo, que está conectada no EZ-KIT. Essa imagem é armazenada na memória para

posterior utilização. A Figura 5.3.26 é a imagem do fundo com o objeto que é capturado pela câmera, que também é armazenada na memória. O processamento dessas duas imagens é feito utilizando-se os parâmetros descritos no capítulo 4. O resultado obtido é mostrado na Figura 5.3.27. O objeto extraído pode ser visualizado na TV que está conectada ao EZ-KIT.

5.4 ALGORITMO PROPOSTO EM YCbCr

Para a implementação no MATLAB, o algoritmo proposto seguiu os passos mostrados na Figura 5.4.28. Começa-se definindo os parâmetros das variáveis de inicialização, tais como: tamanho e quantidade de quadros. Capturam-se as imagens de referência. Logo em seguida são calculados a média e o desvio padrão desses quadros. Para determinação do limiar, o programa utiliza o maior e menor desvio padrão de cada *pixel* das imagens de referência, bem como a amplitude máxima e mínima dos *pixels* dessas imagens. Quando a imagem com o objeto é capturado, é feita uma comparação com a imagem média do fundo, tendo como parâmetros de ponderação os desvios padrões e as amplitudes calculadas. Essas comparações decidem quem é o objeto da imagem.

O algoritmo proposto foi implementado no MATLAB para validação da proposta, levando em consideração as restrições relativas a uma futura implementação em *hardware*. Os resultados obtidos foram comparados com Printes [17], pois seus resultados foram primeiramente simulados no MATLAB. Nessa comparação, foram usadas as seqüências de imagens utilizadas no trabalho de Printes [17].

Para comparar a performance dos dois algoritmos, foi utilizado um microcomputador com sistema operacional Windows XP, processador AMD XP 2400+ e memória RAM 256 Mb de RAM, com o MATLAB 7.0 instalado.



Figura 5.3.25: Imagem de um fundo, capturado pela câmera.



Figura 5.3.26: Imagem do fundo com objeto, capturado pela câmera.



Figura 5.3.27: Extração do objeto visualizado pela TV

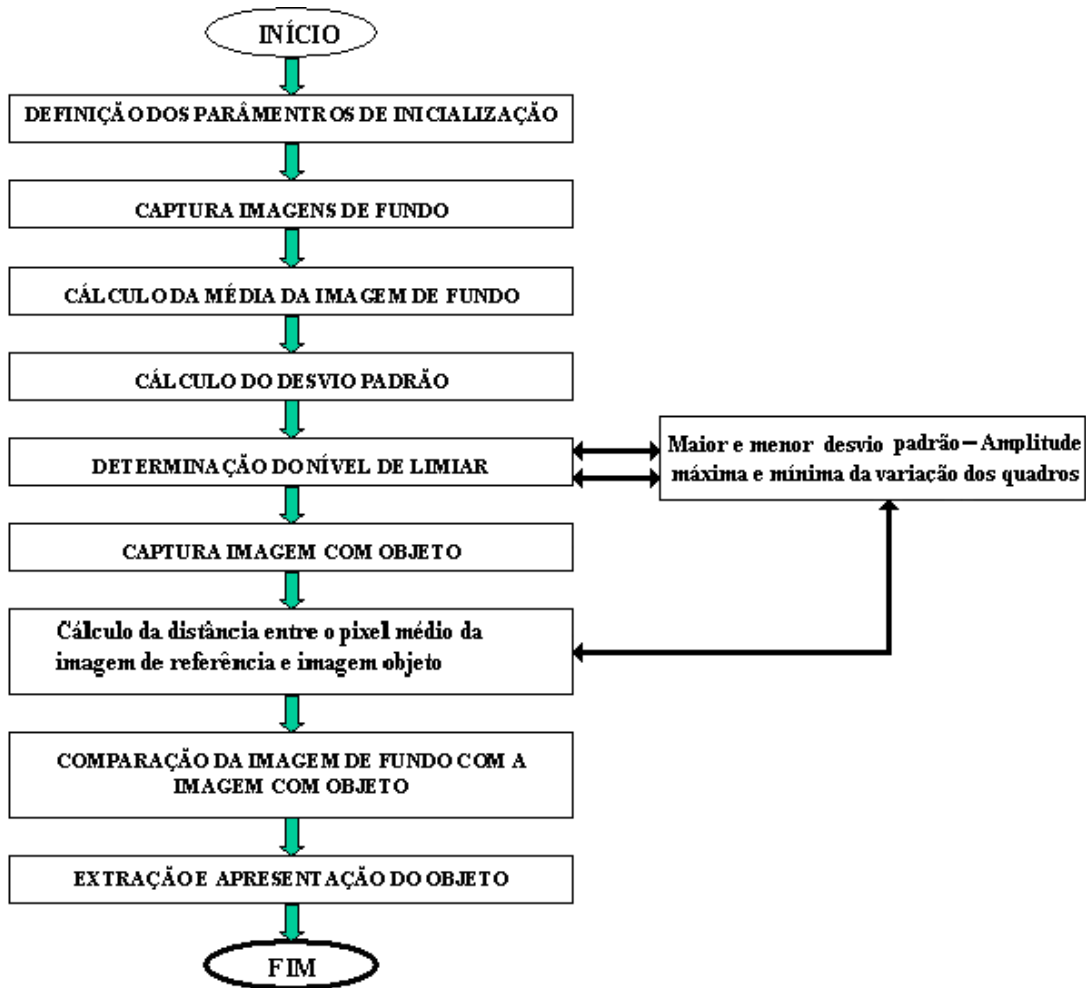


Figura 5.4.28: Diagrama do algoritmo proposto implementado no MATLAB.

O código do algoritmo de Printes [17] tem como base o sistema de cores RGB e levou aproximadamente dois minutos para ser executado por completo. O algoritmo proposto nesta dissertação teve uma performance muito superior, levou aproximadamente somente 12 segundos para a sua execução por completo. Essa diferença de performance foi conseguida devido a algumas tarefas executadas no algoritmo antigo (como cálculos do histograma das imagens) que não foram necessárias no novo algoritmo. Outro fator que justifica tamanho ganho em performance é que o novo algoritmo evita executar cálculos complexos que demandam muito tempo computacional.

Na Figura 5.4.29 são apresentados os resultados obtidos desta simulação. Foram usadas seqüências de imagens com 30 quadros. As imagens foram submetidas ao mesmo processo de extração de objeto descrito neste documento e implementadas no MATLAB 7.0, do qual são extraídas as Figuras e mostrado o resultado final.

Na Figura 5.4.29a tem-se a média da imagem de referência e na Figura 5.4.29b a imagem com o objeto que deverá ser extraído. A Figura 5.4.29c representa o resultado da extração do objeto utilizando o algoritmo de Horpraset [8]. Note que os ruídos que aparecem nessa imagem são distribuídos em quase toda imagem, o próprio objeto apresenta alguma falhas de detecção. A morfologia da imagem foi bastante preservada. Na Figura 5.4.29d é mostrada a imagem extraída utilizando-se o algoritmo proposto. Note que o ruído é menor que na imagem anterior, embora mais concentrado em algumas partes da imagem, os problemas de detecção de sombra e iluminação neste exemplo são muito mais visíveis. Um outro fato significativo é que, neste algoritmo, o contorno da imagem foi um pouco modificado, mas sem comprometer a forma do objeto.

Portanto, embora a imagem extraída pelo novo algoritmo sofra algumas perdas em relação à outra imagem, seu resultado é bastante satisfatório, para aplicações que exijam processamento em tempo real.

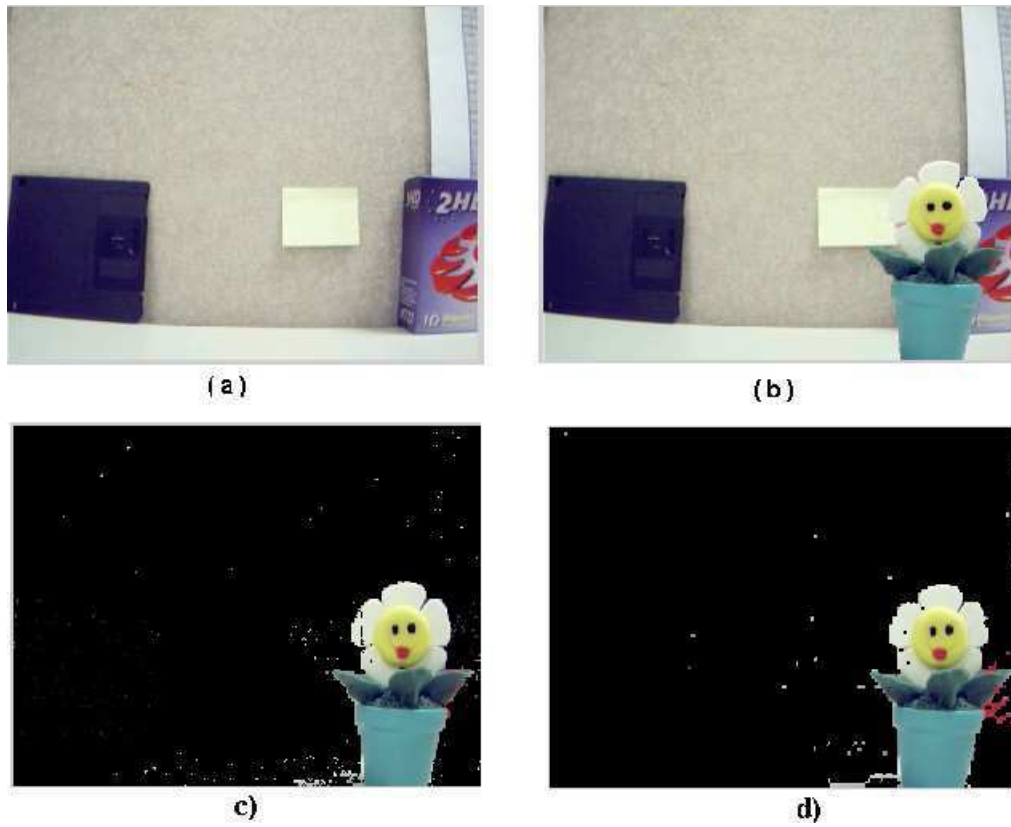


Figura 5.4.29: Diagrama do algoritmo proposto implementado no MATLAB: a) Imagem de referência; b) Imagem com objeto; c) Imagem extraída com algoritmo de Horprasert; d) Imagem extraída com o algoritmo proposto.

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.

CONCLUSÃO E SUGESTÕES

6.1 ANÁLISE DOS RESULTADOS

Após o desenvolvimento desta dissertação, pode-se chegar a duas conclusões importantes:

1 – A implementação do algoritmo de Horpraset [8] na plataforma de desenvolvimento (*EZ-KIT LITE ADSP BF-533 BLACKFIN*) não foi satisfatória, devido à visualização do objeto em tempo real não ser possível, utilizando-se somente os recursos disponíveis na placa. Ela apresenta limitações de *hardware* (tais como: uma única PPI para chavear o *decoder* e *encoder* de vídeo) que não permitem fazer a aquisição, processamento e amostragem da imagem em tempo real.

Outro fato constatado foi que a codificação do algoritmo implementado nessa plataforma ficou muito extensa, devido ao método de extração utilizado neste trabalho ser complexo e utilizar o plano de cores RGB, sendo necessárias as conversões de sistemas de cores. Conseqüentemente, o tempo computacional utilizado pelo sistema foi extremamente grande, não permitindo, assim, a execução do programa em tempo real. Portanto, o algoritmo de Horpraset [8] é complexo e exige um esforço computacional de que a plataforma utilizada não dispõe.

2 – O algoritmo proposto no capítulo 5 é simples e eficiente (comparado com o de Horpraset [8]) em aplicações que dispensam precisão na silhueta do objeto. Devido à sua simplicidade computacional, sinaliza como uma ótima opção para aplicação que exige processamento em tempo real, desde que o *encoder* e *decoder* de vídeo utilizem o padrão de cores YCbCr.

6.2 TRABALHOS FUTUROS

O objetivo principal deste trabalho foi estabelecer um novo método de extração de imagens baseado em trabalhos anteriores. As propostas de trabalhos futuros são focadas no método proposto no capítulo 5.

O algoritmo proposto utiliza a métrica dos máximos para calcular a distância entre os *pixels*. Sugere-se a substituição dessa métrica pela métrica da soma e a avaliação dos resultados obtidos.

Implementar em uma plataforma DSP o algoritmo proposto. Essa implementação abre perspectivas para se pesquisar e implementar aplicações como:

- Construir sistemas de interação com o computador, a partir da análise de faces, sem precisar equipamentos conectados a ele.

- Reconhecimento de padrões.

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.

REFERÊNCIAS E
OBRAS
CONSULTADAS

-
- [1] ANALOG DEVICE: Disponível em: <http://www.analog.com/>. Acesso em 01 de outubro de 2004
- [2] BRONGGI, Alberto. *Parallel and Local Feature Extraction: A Real-Time Approach to Road Boundary Detection*. IEEE Transaction on Imagem Processing, 1995.
- [3] BUTLER, D.; SRIDHARAM, S.; BOVE, M. *Real-Time Adaptive Background Segmentation*. MIT Media Laboratory, Queensland University of Technology, 2000.
- [4] CHAI, Douglas; BOUZERDOUM, Abdesselam. *A Bayesian Approach to Skin Color Cassification in YcbCr Color Space*. School of Engineering and Mathematics Edith Cowan University, 2000
- [5] GONZALES, Raphael C., WOODS, Richard C.. *Processamento Digital de Imagens*. Traduzido por Roberto Marcondes Cesar Junior e Luciano da Fontoura Costa. São Paulo: Editora Edgard Blücher Ltda. Tradução de: Digital Imagem Processing, 2000.
- [6] HARVILLE, M.; GORDON, G.; WOODFILL, J. *Foreground Segmentation Using Adaptive Mixture Models in Color and Depth*. Detection and Recognition of Events in Video, 2001, Proceedings IEEE Workshop on, 2001, pp. 3-11.
- [7] HONG, Dongpyo; WOO, Woontack. *A Brackground Subtraction for a Vision-Base User Interface*. In International Conference on Imagem Processing-ICIP/IEEE, 2003.
- [8] HORPRASERT, Thanarat; HARWOOD, David; DAVIS, Larry S. *A statistical approach for Real Time Robust background subtraction*. In IEEE ICCV'99 FRAME-RATE WORKSHOP, 1999.
- [9] IVANOV, Y.; ROBICK, A.; LIU, J. *Fast lighting independent background subtraction*. International Journal of Computer Vision, 372, pp.199-209, June 2000.
- [10] KIM, Hyun Mun; KIM, Woo-Shik; CHO, Dae-Sung. *A new color transform for RGB coding*. In International Conference on Imagem Processing-ICIP/IEEE, 2004.
- [11] LEE, Ping Hsin; SRINIVASAN, Vivek; SUNDARAJAM, Arvind. *Face Detection*. IEEE Transaction on Imagem Processing, 2001.
- [12] LI, Dalong. *Moving Objects Detection By Block Comparison*. Electronics, Circuit and Systems, 2000. ICECS 2000. The 7th IEEE International Conference On, Volume 1, 2000.
- [13] LOPES, J. M. Brisson. *Teoria das cores*: Disponível em <http://www.mat.ufmg.br/>. Acesso em 12 de setembro de 2004.
- [14] MADS, Lindstrøm, MOHAMAD, Zind. *Model-Based Background Subtraction System*. Master Thesis, Aalborg University, Esbjerg - Janeiro de 2004.
- [15] MATLAB: Disponível em: <http://www.mathworks.com/>. Acesso em 10 de setembro de 2004.

- [16] PARENTE, Jozias de Oliveira. *Implementação em Hardware de um Sistema para Extração de Objetos de um Fundo Não Homogêneo em Tempo Real*. Dissertação de Mestrado, Universidade Federal de Campina Grande, Setembro 2003.
- [17] PRINTES, André Luiz. *Circuito Integrado para Extração de fundo não homogêneo de imagens dinâmicas em tempo real*: Dissertação de Mestrado, Universidade Federal de Campina Grande, Dezembro 2002.
- [18] ROSIN, Paul L. *Thresholding for Change Detection*. In sixth International Conference Computer Vision, 1998.
- [19] SANTOS, Vivian Toledo, *Segmentação de imagens mamográficas para detecção de nódulos em mamas densas*. Dissertação de Mestrado, Universidade de São Carlos, Novembro 2002.
- [20] THONGKAMWITOOM, T; ARAMVITH, S.; CHALIDABHONGSE, T. H. *An Adaptive Real-Time Background Subtraction and Moving Shadow Detection*. In Ieee International Conference on Multimedia And Expo (ICME), pp. 1459-1462, 2004.
- [21] TIEHAN, Lv; OZER, Burak; WOLF, Wayne. *A Real-Time Background Subtraction Method with Camera Motion Compensation*. In IEE International Conference on Multimedia And Expo (ICME), 2004.
- [22] TONIETTO, L. *Análise de algoritmos para Chroma-key*. Disponível em: <http://www.inf.unisinos.br/~marcelow/ensino/tc/ckey/ckey.html>. Acesso em: 23 de abril de 2004.
- [23] TOYAMA, K; *et al. Wallflower: Principles and Practice of Background Maintenance*. In International Conference on Computer Vision, Kerkira, Greece, pp. 255-261, 1999.
- [24] VERILOG. Disponível em: <http://www.verilog.com>. Acesso em 1 de outubro de 2004.
- [25] WU, Q.; JENG, B., *Background subtraction based on logarithmicintensities*. IEEE Pattern Recognition Letters, 2002.
- [26] H., Ismail, H.,David, D.S., Larry, “*W^A: Real-Time Surveillance of People and Their Activities*”. Disponível em: <http://www.umiacs.umd.edu/users/hismail/Publications/Proposal.pdf>. Acesso em 19 de janeiro de 2003.
- [27] CHAI, Douglas; BOUZENLOUM, Abdesselam. *A Bayesian Approach to Skin Color Classification in YcbCr Color Space*. IEEE Transaction on Image Processing, Australia, pp. 421-424, 2000.
- [28] *The perfect color converter*: Disponível em: <http://www.bitjazz.com/sheervideo>

- [29] H., Ismail, H., David, D.S., Larry, “*W^A: Real-Time Surveillance of People and Their Activities*”. Disponível em:
<http://www.umiacs.umd.edu/users/hisail/Publications/Proposal.pdf>. Acesso em 19 de janeiro de 2003.
- [30] LIMA, Elon Lages. Curso de Análise, Vol. 2. Instituto de Matemática Pura e Aplicada. Rio de Janeiro. Editora LTC Ltda. 1995.
- [31] BARBETA, Pedro A.; REIS, Marcelo M.; *Estatística para Curso de Engenharia e Informática*. São Paulo: Editora Atlas S.A. 2004.

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.

APÊNDICE

Arquitetura do *Hardware* do *ADSP-BF533 EZ-KIT Lite* para Processamento de Imagens Digitais

Histórico de Revisões:

REV	Autor	Data	Descrição
A	FGN	02/05/2006	Versão Inicial

Introdução

Este documento tem como finalidade a descrição da arquitetura do *hardware* e a apresentação da estrutura do *software* adotados para implementar um sistema de extração de objetos utilizando o processador embarcado *Blackfin ADSP-BF533* na plataforma de desenvolvimento *ADSP-BF533 EZ-KIT Lite*. Os testes práticos e os resultados obtidos são demonstrados em um documento complementar a este: “**Procedimento de teste.doc**”.

Especificação

As especificações iniciais do sistema são:

- Tempo real de processamento;
- Utilizar na entrada do sistema um sinal proveniente de uma câmera de vídeo analógica de padrão NTSC que possua uma resolução de quadro de 320x240;
- Realizar o aprendizado de uma imagem de fundo estática utilizando 64 quadros;
- Extrair o objeto em movimento que adentrar no fundo através do algoritmo desenvolvido pelo proponente do projeto;
- Visualizar o objeto extraído em um monitor de TV, deixando o fundo totalmente escuro.

Após a avaliação da arquitetura e estudo das possibilidades de implementação, as especificações finais do sistema são:

- Tempo real de processamento;
Solução: Não é possível atingir o tempo real de processamento devido à limitações no *hardware* do *EZ-KIT Lite*.
- Utilizar na entrada do sistema um sinal proveniente de uma câmera de vídeo analógica de padrão NTSC que possua uma resolução de quadro de 720x525;
Solução: A câmera analógica utilizada possui padrão NTSC com uma resolução de quadro de 720x525 ao invés de 320x240.
- Realizar o aprendizado de uma imagem de fundo estática utilizando 5 quadros;
Solução: Para a transformação analógico-digital dos quadros, será utilizada a entrada analógica AVIN1 do *video decoder ADV7183*. São utilizados 5 quadros ao invés de 64 para otimizar o desempenho do sistema.
- Extrair o objeto estático através do algoritmo desenvolvido pelo proponente do projeto;
Solução: Após a digitalização e o aprendizado, o processador realiza os cálculos do algoritmo proposto para a extração do objeto. Para isso, o algoritmo foi

implementado mesclando linguagem C e linguagem *Assembly* do processador embarcado.

- Visualizar o objeto extraído estaticamente em um monitor de TV após a realização dos cálculos do algoritmo, deixando o fundo totalmente escuro.

Solução: Após a execução do algoritmo e a transformação digital-analógica, será utilizada a saída analógica DAC_B do *video encoder* ADV7171 para conexão com o monitor de TV.

Descrição do *Hardware*

Para o desenvolvimento do sistema de extração de objetos, a arquitetura de *hardware* disponível no *ADSP-BF533 EZ-KIT Lite* utiliza os seguintes componentes:

- **Processador Embarcado ADSP-BF533:** O núcleo do processador tem capacidade de operar em uma frequência de até 750MHz. Essa frequência é denominada *core clock* e é gerada por uma PLL interna a partir de um oscilador de 27MHz. A *EBIU (External Bus Interface Unit)* serve de interface com memórias externas e suporta tanto memórias síncronas (*SDRAM*) de acordo com o padrão PC100 ou PC133 quanto memórias assíncronas (*SRAM, FLASH, etc.*) através de 4 bancos de até 1MByte. São utilizadas as seguintes interfaces do processador: a *PPI (Parallel Port Interface)*, o canal *DMA0 (Direct Memory Access)* e os *Flags* Programáveis.

- **Memória SDRAM (Synchronous Dynamic RAM):** Está disponível no *EZ-KIT Lite* uma memória *SDRAM* com 32MBytes de densidade totalmente adequada à projetos com imagens digitais. Esta opera na frequência do sistema com taxa máxima até 133MHz. Essa frequência é denominada *system clock* e é gerada pelo processador para o sistema a partir de um oscilador externo de 27MHz.

- **Memória FLASH:** O sistema utiliza o *Port A* da memória *FLASH* através de I/Os mapeados em memória com o propósito de realizar o link do *clock* do *Video decoder* ADV7183 na interface *PPI* do processador e para retirar o componente do estado de *Reset*. A *Port A* da *FLASH* também é utilizada para retirar o *Video Encoder* ADV7171 do estado de *Reset*.

- **Video decoder ADV7183:** O ADV7183 utiliza uma entrada analógica AVIN1 para receber o sinal de vídeo composto no padrão NTSC. É possível configurar o modo *Active data*, onde somente os dados digitais ativos da imagem são transferidos para o processador via interface *PPI* e todos os sinais de sincronismo são descartados.

- **Video decoder ADV7171:** Depois que todos os cálculos foram concluídos, os dados digitais ativos da imagem resultante são montados no esqueleto do padrão digital NTSC e enviado via interface *PPI* para o ADV7171. Utilizando a saída DAC_B do componente, o sinal de vídeo composto analógico é inserido na entrada de vídeo do monitor de TV.

O princípio de operação do *hardware* utilizado no projeto do sistema de extração é ilustrado no diagrama em blocos da figura 1.

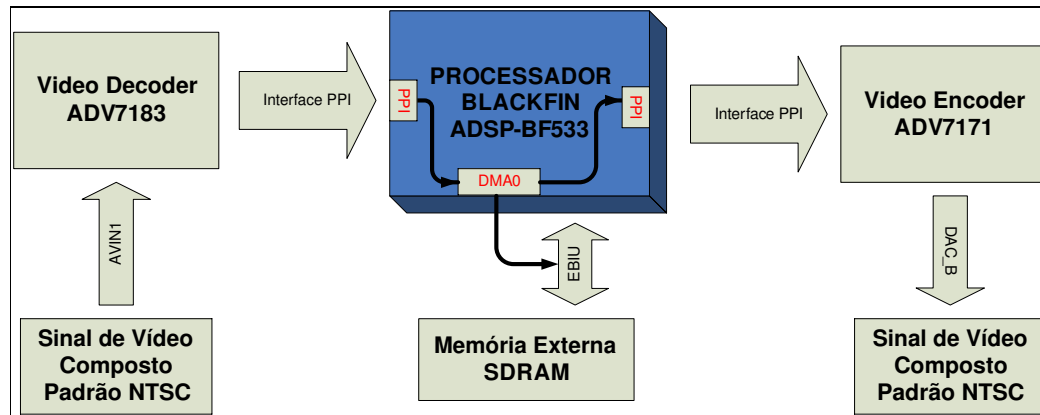


Figura 1: Diagrama em blocos da utilização do *hardware* disponível no kit de desenvolvimento do processador *Blackfin ADSP-BF533*.

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.

ANEXO

RECOMMENDATION ITU-R BT.656-4

**INTERFACES FOR DIGITAL COMPONENT VIDEO SIGNALS IN 525-LINE
AND 625-LINE TELEVISION SYSTEMS OPERATING AT THE 4:2:2
LEVEL OF RECOMMENDATION ITU-R BT.601 (PART A)**

(Question ITU-R 65/11)

(1986-1992-1994-1995-1998)

The ITU Radiocommunication Assembly,

considering

- a) that there are clear advantages for television broadcasting organizations and programme producers in digital studio standards which have the greatest number of significant parameter values common to 525-line and 625-line systems;
- b) that a worldwide compatible digital approach will permit the development of equipment with many common features, permit operating economies and facilitate the international exchange of programmes;
- c) that to implement the above objectives, agreement has been reached on the fundamental encoding parameters of digital television for studios in the form of Recommendation ITU-R BT.601;
- d) that the practical implementation of Recommendation ITU-R BT.601 requires definition of details of interfaces and the data streams traversing them;
- e) that such interfaces should have a maximum of commonality between 525-line and 625-line versions;
- f) that in the practical implementation of Recommendation ITU-R BT.601 it is desirable that interfaces be defined in both serial and parallel forms;
- g) that digital television signals produced by these interfaces may be a potential source of interference to other services, and due notice must be taken of No. 964 of the Radio Regulations (RR),

recommends

that where interfaces are required for component-coded digital video signals described in Recommendation ITU-R BT.601 (Part A) in television studios, the interfaces and the data streams that will traverse them should be in accordance with the following description, defining both bit-parallel and bit-serial implementations.

1 Introduction

This Recommendation describes the means of interconnecting digital television equipment operating on the 525-line or 625-line standards and complying with the 4:2:2 encoding parameters as defined in Recommendation ITU-R BT.601 (Part A).

Part 1 describes the signal format common to both interfaces.

Part 2 describes the particular characteristics of the bit-parallel interface.

Part 3 describes the particular characteristics of the bit-serial interface.

Supplementary information is to be found in Annex 1.

PART 1

Common signal format of the interfaces

1 General description of the interfaces

The interfaces provide a unidirectional interconnection between a single source and a single destination.

A signal format common to both parallel and serial interfaces is described in § 2.

The data signals are in the form of binary information coded in 8-bit or, optionally, 10-bit words (see Note 1). These signals are:

- video signals,
- timing reference signals,
- ancillary signals.

NOTE 1 – Within this Recommendation, the contents of digital words are expressed in both decimal and hexadecimal form. To avoid confusion between 8-bit and 10-bit representations, the eight most significant bits are considered to be an integer part while the two additional bits, if present, are considered to be fractional parts.

For example, the bit pattern 10010001 would be expressed as 145_d or 91_h, whereas the pattern 1001000101 is expressed as 145.25_d or 91.4_h.

Where no fractional part is shown, it should be assumed to have the binary value 00.

Eight-bit words occupy the left most significant bits of a 10-bit word, i.e. bit 9 to bit 2, where bit 9 is the most significant bit.

2 Video data

2.1 CODING CHARACTERISTICS

The video data is in compliance with Recommendation ITU-R BT.601 Part A, and with the field-blanking definition shown in Table 1.

TABLE 1
Field interval definitions

		625	525
V-digital field blanking			
Field 1	Start (V = 1)	Line 624	Line 1
	Finish (V = 0)	Line 23	Line 20
Field 2	Start (V = 1)	Line 311	Line 264
	Finish (V = 0)	Line 336	Line 283
F-digital field identification			
Field 1	F = 0	Line 1	Line 4
Field 2	F = 1	Line 313	Line 266

NOTE 1 – Signals F and V change state synchronously with the end of active video timing reference code at the beginning of the digital line.

NOTE 2 – Definition of line numbers is to be found in Recommendation ITU-R BT.470. Note that digital line number changes state prior to O_H as described in Recommendation ITU-R BT.601 (Part A).

NOTE 3 – Designers should be aware that the “1” to “0” transition of the V-bit may not necessarily occur on line 20 (283) in some equipment conforming to previous versions of this Recommendation for 525-line signals.

2.2 VIDEO DATA FORMAT

The data words in which the eight most significant bits are all set to 1 or are all set to 0 are reserved for data identification purposes and consequently only 254 of the possible 256 8-bit words (or 1 016 of the possible 1 024 10-bit words) may be used to express a signal value.

The video data words are conveyed as a 27 Mword/s multiplex in the following order:

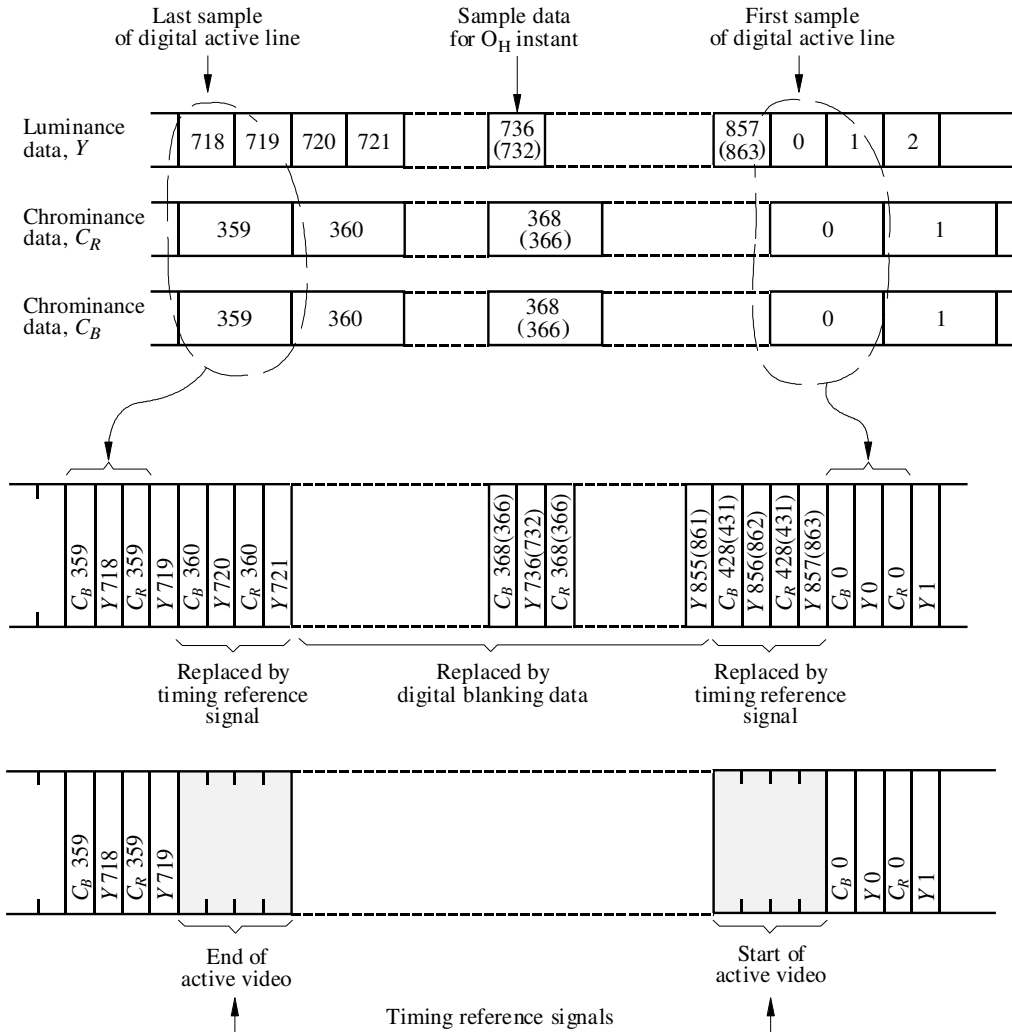
$$C_B, Y, C_R, Y, C_B, Y, C_R, \text{ etc.}$$

where the word sequence C_B, Y, C_R , refers to co-sited luminance and colour-difference samples and the following word, Y , corresponds to the next luminance sample.

2.3 INTERFACE SIGNAL STRUCTURE

Figure 1 shows the ways in which the video sample data is incorporated in the interface data stream. Sample identification in Fig. 1 is in accordance with the identification in Recommendation ITU-R BT.601 (Part A).

FIGURE 1
Composition of interface data stream



Note 1 – Sample identification numbers in parentheses are for 625-line systems where these differ from those for 525-line systems. (See also Recommendation ITU-R BT.803.)

D01

2.4 VIDEO TIMING REFERENCE CODES (SAV, EAV)

There are two timing reference signals, one at the beginning of each video data block (start of active video, SAV) and one at the end of each video data block (end of active video, EAV) as shown in Fig. 1.

Each timing reference signal consists of a four word sequence in the following format: FF 00 00 XY. (Values are expressed in hexadecimal notation. FF 00 values are reserved for use in timing reference signals.) The first three words are a fixed preamble. The fourth word contains information defining field 2 identification, the state of field blanking, and the state of line blanking. The assignment of bits within the timing reference signal is shown in Table 2.

TABLE 2

Video timing reference codes

Data bit number	First word (FF)	Second word (00)	Third word (00)	Fourth word (XY)
9 (MSB)	1	0	0	1
8	1	0	0	F
7	1	0	0	V
6	1	0	0	H
5	1	0	0	P ₃
4	1	0	0	P ₂
3	1	0	0	P ₁
2	1	0	0	P ₀
1 (Note 2)	1	0	0	0
0	1	0	0	0

NOTE 1 – The values shown are those recommended for 10-bit interfaces.

NOTE 2 – For compatibility with existing 8-bit interfaces, the values of bits D₁ and D₀ are not defined.

F = 0 during field 1, 1 during field 2

V = 0 elsewhere, 1 during field blanking

H = 0 in SAV, 1 in EAV

P₀, P₁, P₂, P₃: protection bits (see Table 3)

MSB: most significant bit

Table 1 defines the state of the V and F bits.

Bits P₀, P₁, P₂, P₃, have states dependent on the states of the bits F, V and H as shown in Table 3. At the receiver this arrangement permits one-bit errors to be corrected and two-bit errors to be detected.

TABLE 3

Protection bits

F	V	H	P ₃	P ₂	P ₁	P ₀
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	1	1
1	0	1	1	0	1	0
1	1	0	1	1	0	0
1	1	1	0	0	0	1

2.5 ANCILLARY DATA

The ancillary signals should comply with Recommendation ITU-R BT.1364.

2.6 DATA WORDS DURING BLANKING

The data words occurring during digital blanking intervals that are not used for the timing reference code or for ancillary data are filled with the sequence 80.0_h, 10.0_h, 80.0_h, 10.0_h

etc. corresponding to the blanking level of the C_B , Y , C_R , Y signals respectively, appropriately placed in the multiplexed data.

PART 2

Bit-parallel interface

1 General description of the interface

The bits of the digital code words that describe the video signal are transmitted in parallel by means of eight (optionally, ten) conductor pairs, where each carries a multiplexed stream of bits (of the same significance) of each of the component signals, C_B , Y , C_R , Y .

The eight pairs also carry ancillary data that is time-multiplexed into the data stream during video blanking intervals. An additional pair provides a synchronous clock at 27 MHz.

The signals on the interface are transmitted using balanced conductor pairs. Cable lengths of up to 50 m (–, ~ 160 feet) without equalization and up to 200 m (–, ~ 650 feet) with appropriate equalization may be employed.

The interconnection employs a twenty-five pin D-subminiature connector equipped with a locking mechanism (see § 5).

For convenience, the bits of the data word are assigned the names DATA 0 to DATA 9.

The entire word is designated as DATA (0-9). DATA 9 is the most significant bit. Eight-bit data words occupy DATA (2-9).

Video data is transmitted in NRZ form in real time (unbuffered) in blocks, each comprising one active television line.

2 Data format

The interface carries data in the form of eight (optionally, ten) parallel data bits and a separate synchronous clock. Data is coded in NRZ form. The recommended data format is described in Part 1.

3 Clock signal

3.1 GENERAL

The clock signal is a 27 MHz square wave where the 0-1 transition represents the data transfer time. This signal has the following characteristics:

Width: 18.5 ± 3 ns

Jitter: Less than 3 ns from the average period over one field.

NOTE 1 – This jitter specification, while appropriate for an effective parallel interface, is not suitable for clocking digital-to-analogue conversion or parallel-to-serial conversion.

3.2 CLOCK-TO-DATA TIMING RELATIONSHIP

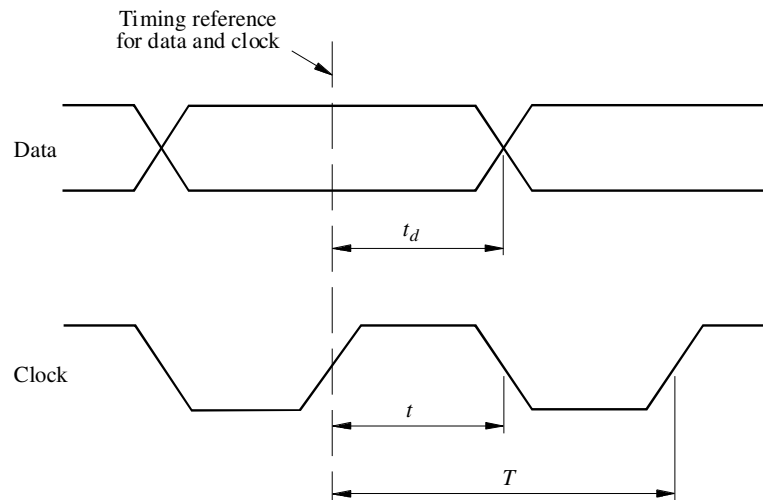
The positive transition of the clock signal shall occur midway between data transitions as shown in Fig. 2.

4 Electrical characteristics of the interface

4.1 GENERAL

Each line driver (source) has a balanced output and the corresponding line receiver (destination) a balanced input (see Fig. 3). Although the use of ECL technology is not specified, the line driver and receiver must be ECL-compatible, i.e. they must permit the use of ECL for either drivers or receivers. All digital signal time intervals are measured between the half-amplitude points.

FIGURE 2
Clock-to-data timing (at source)



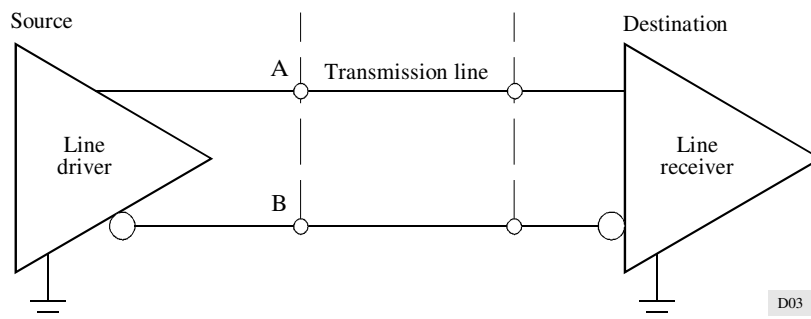
Clock period (625): $T = \frac{1}{1\,728 f_H} = 37 \text{ ns}$
 Clock period (525): $T = \frac{1}{1\,716 f_H} = 37 \text{ ns}$
 Clock pulse width: $t = 18.5 \pm 3 \text{ ns}$
 Data timing – sending end: $t_d = 18.5 \pm 3 \text{ ns}$
 f_H : line frequency

D02

4.2 LOGIC CONVENTION

The A terminal of the line driver is positive with respect to the B terminal for a binary 1 and negative for a binary 0 (see Fig. 3).

FIGURE 3
Line driver and line receiver interconnection



D03

4.3 LINE DRIVER CHARACTERISTICS (*SOURCE*)

4.3.1 *Output impedance*: 110 Ω maximum.

4.3.2 *Common mode voltage*: $-1.29 \text{ V} \pm 15\%$ (both terminals relative to ground).

4.3.3 *Signal amplitude*: 0.8 to 2.0 V peak-to-peak, measured across a 110 Ω resistive load.

4.3.4 *Rise and fall times*: less than 5 ns, measured between the 20% and 80% amplitude points, with a 110 Ω resistive load. The difference between rise and fall times must not exceed 2 ns.

4.4 LINE RECEIVER CHARACTERISTICS (*DESTINATION*)

4.4.1 *Input impedance*: 110 $\Omega \pm 10 \Omega$.

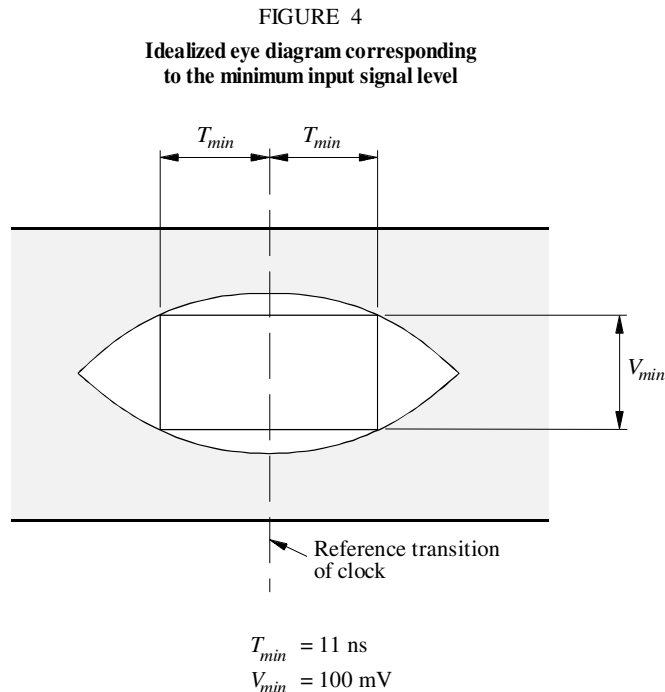
4.4.2 *Maximum input signal*: 2.0 V peak-to-peak.

4.4.3 *Minimum input signal*: 185 mV peak-to-peak.

However, the line receiver must sense correctly the binary data when a random data signal produces the conditions represented by the eye diagram in Fig. 4 at the data detection point.

4.4.4 *Maximum common mode signal*: $\pm 0.5 \text{ V}$, comprising interference in the range 0 to 15 kHz (both terminals to ground).

4.4.5 *Differential delay*: Data must be correctly sensed when the clock-to-data differential delay is in the range between $\pm 11 \text{ ns}$ (see Fig. 4).



Note 1 – The width of the window in the eye diagram, within which data must be correctly detected comprises $\pm 3 \text{ ns}$ clock jitter, $\pm 3 \text{ ns}$ data timing (see § 3.2), $\pm 5 \text{ ns}$ available for differences in delay between pairs of the cable. (See also Recommendation ITU-R BT.803.)

5 Mechanical details of the connector

The interface uses the 25 contact type D subminiature connector specified in ISO Doc. 2110-1980, with the contact assignment shown in Table 4.

TABLE 4
Contact assignments

Contact	Signal line
1	Clock
2	System ground A
3	Data 9 (MSB)
4	Data 8
5	Data 7
6	Data 6
7	Data 5
8	Data 4
9	Data 3
10	Data 2
11	Data 1
12	Data 0
13	Cable shield
14	Clock return
15	System ground B
16	Data 9 return
17	Data 8 return
18	Data 7 return
19	Data 6 return
20	Data 5 return
21	Data 4 return
22	Data 3 return
23	Data 2 return
24	Data 1 return
25	Data 0 return

NOTE 1 – The cable shield (contact 13) is for the purpose of controlling electromagnetic radiation from the cable. It is recommended that contact 13 should provide high-frequency continuity to the chassis ground at both ends and, in addition, provide DC continuity to the chassis ground at the sending end. (See also Recommendation ITU-R BT.803.)

Connectors are locked together by two UNC 4-40 screws on the cable connectors, which go in female screw locks mounted on the equipment connector. Cable connectors employ pin contacts and equipment connectors employ socket contacts. Shielding of the interconnecting cable and its connectors must be employed (see Note 1).

NOTE 1 – It should be noted that the ninth and eighteenth harmonics of the 13.5 MHz sampling frequency (nominal value) specified in Recommendation ITU-R BT.601 (Part A) fall at the 121.5 and 243 MHz aeronautical emergency channels. Appropriate precautions must therefore be taken in the design and operation of interfaces to ensure that no interference is caused at these frequencies. Emission levels for related equipment are given in CISPR Recommendation: “Information technology equipment – limits of interference and measuring methods”, Doc. CISPR/B (Central Office) 16. Nevertheless, RR No. 964 prohibits any harmful interference on the emergency frequencies. (See also Recommendation ITU-R BT.803.)

PART 3

Bit-serial interface**1 General description of the interface**

The multiplexed data stream of 10-bit words (as described in Part 1) is transmitted over a single channel in bit-serial form. Prior to transmission, additional coding takes place to provide spectral shaping, word synchronization and to facilitate clock recovery (see Note 1).

NOTE 1 – Previous versions of this Recommendation have described a serial interface based on an 8B9B word-mapping technique. Due to implementation difficulties this technique is no longer recommended.

In addition to the 10-bit interface based on scrambling described in this revision of the Recommendation, there exists an 11-bit word format (10B1C) in which the eleventh bit is the complement of the least significant bit (LSB) of the scrambled data word.

2 Coding

The uncoded serial bit-stream is scrambled using the generator polynomial $G1(x) \times G2(x)$, where:

$$\begin{array}{ll} G1(x) = x^9 + x^4 + 1 & \text{to produce a scrambled NRZ signal, and} \\ G2(x) = x + 1 & \text{to produce a polarity-free NRZI sequence.} \end{array}$$

3 Order of transmission

The least significant bit of each 10-bit word shall be transmitted first.

4 Logic convention

The signal is transmitted in NRZI form, for which the bit polarity is irrelevant.

5 Transmission medium

The bit-serial data stream can be conveyed using either a coaxial cable (see § 6) or fibre-optic bearer (see § 7).

6 Characteristics of the electrical interface**6.1 LINE DRIVER CHARACTERISTICS (SOURCE)****6.1.1 Output impedance**

The line driver has an unbalanced output with a source impedance of 75Ω and a return loss of at least 15 dB over a frequency range of 5-270 MHz.

6.1.2 Signal amplitude

The peak-to-peak signal amplitude lies between $800 \text{ mV} \pm 10\%$ measured across a 75Ω resistive load directly connected to the output terminals without any transmission line.

6.1.3 d.c. offset

The d.c. offset with reference to the mid-amplitude point of the signal lies between + 0.5 and - 0.5 V.

6.1.4 Rise and fall times

The rise and fall times, determined between the 20% and 80% amplitude points and measured across a 75 Ω resistive load connected directly to the output terminals, shall lie between 0.75 and 1.50 ns and shall not differ by more than 0.50 ns.

6.1.5 Jitter

The output jitter is specified as follows:

Output jitter (see Note 1)

f_1	= 10 Hz
f_3	= 100 kHz
f_4	= 1/10 of the clock rate
A_1	= 0.2 UI (UI; unit interval) (see Note 2)
A_2	= 0.2 UI

NOTE 1 – 1 UI and 0.2 UI correspond to 3.7 ns and 0.74 ns.

Specification of jitter and jitter measurements methods shall comply with Recommendation ITU-R BT.1363 (Jitter specifications and jitter measurement methods of bit-serial signals conforming to Recommendations ITU-R BT.656, ITU-R BT.799 and ITU-R BT.1120).

NOTE 2 – 0.2 UI for timing jitter is often used in other specifications. There are considerations in specifying 1 UI for timing jitter.

6.2 LINE RECEIVER CHARACTERISTICS (*DESTINATION*)

6.2.1 Terminating impedance

The cable is terminated by 75 Ω with a return loss of at least 15 dB over a frequency range of 5-270 MHz.

6.2.2 Receiver sensitivity (*see Note 1*)

The line receiver must sense correctly random binary data when either connected to a line driver operating at the extreme voltage limits permitted by § 6.1.2 or when connected via a cable having a loss of 40 dB at 270 MHz and a loss characteristic of $1 / \sqrt{f}$.

NOTE 1 – Parameters defined in § 6.1.5, 6.2.2 and 6.2.3 are target values and may be refined in the future with regard to practical implementations of the system.

6.2.3 Interference rejection (*see Note 1*)

When connected directly to a line driver operating at the lower limit specified in § 6.1.2, the line receiver must sense correctly the binary data in the presence of a superimposed interfering signal at the following levels:

d.c.	± 2.5 V
Below 1 kHz:	2.5 V peak-to-peak
1 kHz to 5 MHz:	100 mV peak-to-peak
Above 5 MHz:	40 mV peak-to-peak

NOTE 1 – Parameters defined in § 6.1.5, 6.2.2 and 6.2.3 are target values and may be refined in the future with regard to practical implementations of the system.

6.2.4 Input jitter

Input jitter tolerances needs to be defined. Input jitter is measured with a short cable (2 m). Specification of jitter and jitter measurements methods shall comply with Recommendation ITU-R BT.1363 (Jitter specifications and jitter measurement methods of bit-serial signals conforming to Recommendations ITU-R BT.656, ITU-R BT.799 and ITU-R BT.1120).

6.3 CABLES AND CONNECTORS

6.3.1 Cable

It is recommended that the cable chosen should meet any relevant national standards on electromagnetic radiation.

NOTE 1 – It should be noted that the ninth and eighteenth harmonics of the 13.5 MHz sampling frequency (nominal value) specified in Recommendation ITU-R BT.601 (Part A) fall at the 121.5 and 243 MHz aeronautical emergency channels. Appropriate precautions must therefore be taken in the design and operation of interfaces to ensure that no interference is caused at these frequencies. Emission levels for related equipment are given in CISPR Recommendation: “Information technology equipment – limits of interference and measuring methods” (Doc. CISPR/B (Central Office) 16). Nevertheless, RR No. 964 prohibits any harmful interference on the emergency frequencies. (See also Recommendation ITU-R BT.803.)

6.3.2 Characteristic impedance

The cable used shall have a nominal characteristic impedance of 75 Ω .

6.3.3 Connector characteristics

The connector shall have mechanical characteristics conforming to the standard BNC type (IEC Publication 169-8), and its electrical characteristics should permit it to be used at frequencies up to 850 MHz in 75 Ω circuits.

7 Characteristics of the optical interface

Specifications for the characteristics of the optical interface should comply with general rules of Recommendation ITU-R BT.1367 (Serial Digital Fiber Transmission Systems for Signals Conforming to Recommendations ITU-R BT.656, ITU-R BT.799 and ITU-R BT.1120).

To make use of this Recommendation the following specifications are necessary:

Rise and fall times	< 1.5ns (20% to 80%)
Output jitter (see Note 1)	$f_1 = 10 \text{ Hz}$
	$f_3 = 100 \text{ kHz}$
	$f_4 = 1/10 \text{ of the clock rate}$
	$A_1 = 0.135 \text{ UI (UI; unit interval)}$
	$A_2 = 0.135 \text{ UI}$

Input jitter needs to be defined. Input jitter is measured with a short cable (2 m).

NOTE 1 - Specification of jitter and jitter measurements methods shall comply with Recommendation ITU-R BT.1363 (Jitter specifications and jitter measurement methods of bit-serial signals conforming to Recommendations ITU-R BT.656, ITU-R BT.799 and ITU-R BT.1120).

ANNEX 1

Notes concerning interfaces for digital video signals in 525-line and 625-line television systems

1 Introduction

This Annex includes supplementary information on subjects not yet fully specified, and indicates studies in which further work is required.

2 Definitions

Interface is a concept involving the specification of the interconnection between two items of equipment or systems. The specification includes the type, quantity and function of the interconnection circuits and the type and form of the signals to be interchanged by these circuits.

A parallel interface is an interface in which the bits of a data word are sent simultaneously via separate channels.

A serial interface is an interface in which the bits of a data word, and successive data words, are sent consecutively via a single channel.

3 Parallel interfaces

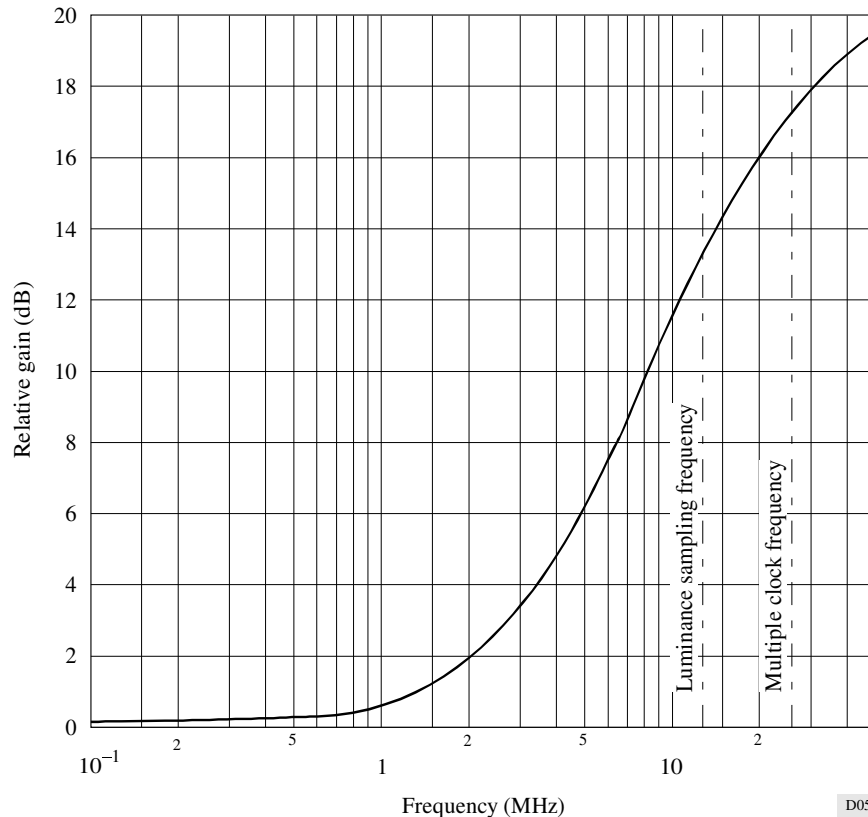
Appropriate coding of the clock signal, such as the use of an alternating parity (AP) coding, has been shown to extend the interconnection distance by reducing the effects of cable attenuation.

To permit correct operation with longer interconnection links, the line receiver may incorporate equalization.

When equalization is used, it may conform to the nominal characteristic of Fig. 5. This characteristic permits operation with a range of cable lengths down to zero. The line receiver must satisfy the maximum input signal condition of § 4.4 of Part 2 of this Recommendation.

FIGURE 5

Line receiver equalization characteristic for small signals



4 Serial interfaces

The transmission of signals can be achieved in both electrical form, using coaxial cable, and in optical form using an optical fibre. Coaxial cables would probably be preferred for connections of medium length, while preference would go to optical fibres for very long connection lengths.

It is possible to implement a system for detection of the occurrence of errors at the receiving end of the connection and thus automatically monitoring its performance. In a fully integrated digital installation or system it may be useful for all interconnections to be transparent to any appropriate digital stream, irrespective of the message content. Thus, although the interface will be used to transmit a video signal, it should be “transparent” to the message content, i.e. it should not base its operation on the known structure of the message itself.

Development work is ongoing on the subject of serial interfaces. In the context of the European Race projects, for example, fibre optic routing systems which can accept a variety of input formats are assembled as part of a pilot installation.

5 Interference with other services

Processing and transmission of digital data, such as digital video signals at high data rates produces a wide spectrum of energy that has the potential to cause cross-talk or interference. In particular, attention is drawn in the present Recommendation to the fact that the ninth and eighteenth harmonics of the 13.5 MHz sampling frequency (nominal value) specified in Recommendation ITU-R BT.601 (Part A) fall at the 121.5 and 243 MHz aeronautical emergency channels. Appropriate precautions must therefore be taken in the design and operation of interfaces to ensure that no interference is caused at these frequencies. Permitted maximum levels of radiated signals from digital data processing equipment are the subject of various national and international standards, and it should be noted that emission levels for such related equipment are given in CISPR

Recommendation: "Information technology equipment – Limits of interference and measuring methods", Doc. CISPR/B (Central Office) 16.

In the case of the bit-parallel interface, work carried out by the Canadian Broadcasting Corporation (CBC) indicates that, with a correct shielding of the cables, no interference problem with other services is to be expected. Radiation levels should comply with the limits given in Table 5. These limits are equivalent to those of the FCC in the United States of America.

TABLE 5

Limits of spurious emissions

Frequency (MHz)	Maximum field strength at 30 m (dB(μ V/m))
30-88	30
88-216	50
216-1 000	70

Transmission by optical fibres eliminates radiation generated by the cable and also prevents conducted common-mode radiation, but the performance of coaxial cable can also be made near-perfect. It is believed that the major portion of any radiation would be from the processing logic and high-power drivers common to both methods. Due to the wideband, random nature of the digital signal, little is gained by frequency optimization.

6 Conclusion

Further studies are required on the practical methods required to ensure acceptably low levels of radiated interference from the digital signals.

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.