

UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Leonardo Dorneles Figueiredo Silva

*Monitoramento do estado de sonolência de motoristas de
automóveis através de análise de imagens de olhos*

São Luís

2012

Leonardo Dorneles Figueiredo Silva

*Monitoramento do estado de sonolência de motoristas de
automóveis através de análise de imagens de olhos*

Dissertação apresentada ao Curso de Engenharia Elétrica da UFMA, como requisito para a obtenção parcial do grau de Mestre em Engenharia Elétrica.

Orientador: Aristófanês Corrêa Silva

Prof. Dr. em Informática - UFMA

Co-orientador: Anselmo Cardoso Paiva

Prof. Dr. em Informática - UFMA

São Luís

2012

Silva, Leonardo Dorneles Figueiredo.

Monitoramento do estado de sonolência de motoristas de automóveis através de análise de imagens de olhos / Leonardo Dorneles Figueiredo Silva – São Luís, 2012.

67 f.

Orientador: Aristófanês Corrêa Silva.

Co-orientador: Anselmo Cardoso Paiva.

Dissertação (Mestrado) – Universidade Federal do Maranhão, Programa de Pós-Graduação em Engenharia Elétrica, 2012.

1. Sistema computacional – Monitoramento dos olhos. 2. Máquina de vetor de suporte. 3. Filtro de Haar. I. Título.

CDU 004.78:611.84

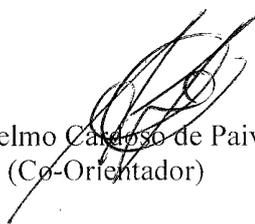
**MONITORAMENTO DO ESTADO DE SONOLÊNCIA
DE MOTORISTAS DE AUTOMÓVEIS ATRAVÉS
DE ANÁLISE DE IMAGENS DE OLHOS**

Leonardo Dorneles Figueiredo Silva

Dissertação aprovada em 28 de fevereiro de 2012.



Prof. Aristófanês Corrêa Silva, Dr.
(Orientador)



Prof. Anselmo Cardoso de Paiva, Dr.
(Co-Orientador)



Profa. Aura Conci, Dra.
(Membro da Banca Examinadora)



Prof. João Viana da Fonseca Neto, Dr.
(Membro da Banca Examinadora)

*Dissertação dedicada ao meu primo, Victor
Figueiredo Prazeres (in memoriam).*

Resumo

O cansaço e a fadiga contribuem para que o condutor de veículos automotores se envolvam em um grande número de acidentes. Esse número poderia ser reduzido caso fosse possível detectar o momento de desatenção e alertar o motorista. Uma metodologia que seja capaz de fazer essa detecção de forma automática deve ser capaz de processar as informações da situação atual do usuário e fornecer a resposta em tempo real de acordo com o seu comportamento e também não atrapalhe o condutor na sua forma natural de dirigir. Neste trabalho foi desenvolvido uma metodologia que utiliza técnicas de processamento de imagens, visão computacional, aprendizado de máquina e características físicas para a detecção da região dos olhos e a análise de seu comportamento com o objetivo de verificar o nível de desatenção de motoristas de automotores.

Palavras-chaves: Monitoramento dos Olhos; Máquina de Vetor de Suporte; Filtro de Haar; Transformada de Hough

Abstract

The tiredness and fatigue contribute to the involvement of drivers in a large number of accidents. This number could be reduced if it was possible to detect the moment of inattention and warn the driver of their condition. A methodology that is able to detect this automatically should be able to process the information of the current status of the driver and provide an advise in real time according to their behavior. However it must not affect the driver in its natural way to drive. In this work was developed a methodology that uses image processing techniques, computer vision, machine learning and physical characteristics to detect the eye region and analysis of their behavior with the objective of verifying the level of inattention of drivers of cars.

Keywords: Eye Tracking, Support Vector Machine, Haar Filters, Hough Transform.

Agradecimentos

À Deus, por permitir frutos dos meus esforços;

À minha família, pelo apoio e incentivo;

À minha namorada Karol, pelo carinho e paciência ;

Ao professor orientador Aristófanês Corrêa Silva, pelas contribuições, correções, conselhos e sobretudo paciência;

Ao professor coorientador Anselmo Cardoso Paiva, pelas contribuições e sugestões de materiais durante a pesquisa;

Aos colegas do LabPAI: Darlan, Stelmo e João Dallyson pelo apoio e contribuições durante a pesquisa;

Aos professores da banca examinadora;

Ao CNPq pelo apoio financeiro concedido durante o mestrado.

A todos que contribuíram, mas que por falta de lembrança não foram citados.

Eu acredito demais na sorte. E tenho constatado que, quanto mais duro eu trabalho, mais sorte eu tenho. (Thomas Jefferson)

Sumário

Lista de Figuras	9
Lista de Tabelas	12
1 Introdução	13
1.1 Trabalhos Relacionados	14
1.2 Organização do trabalho	16
2 Fundamentação Teórica	17
2.1 Pré-processamento	17
2.1.1 Transformações em Níveis de Cinza	17
2.1.1.1 Especificação do Histograma	18
2.1.2 Transformações Utilizando Posição do <i>pixel</i>	19
2.1.2.1 Erro Médio Absoluto (EMA)	20
2.2 Pré-Processamento Local	20
2.2.1 Suavizadores	20
2.2.1.1 Suavização Gaussiana	20
2.2.2 Operadores de Gradiente	21
2.2.2.1 Detector de Bordas Sobel	23
2.3 Transformações Geométricas	23
2.3.1 Translação	23
2.3.2 Escala	24
2.3.3 Rotação	24
2.3.4 Espelhamento	25

2.3.5	Coordenadas homogêneas	25
2.4	Variação de Iluminação	26
2.4.1	Extração de Características Invariantes à Iluminação	27
2.4.2	Formas Canônica	27
2.4.2.1	Single Scale Retinex	27
2.4.3	Normalização da Cor	28
2.5	Segmentação	28
2.5.1	Segmentação Baseada na Cor da Pele	29
2.5.2	Segmentação Baseada em bordas	30
2.5.2.1	Transformada de Hough	31
2.6	Extração de Atributos	35
2.6.1	Transformadas de Wavelets	36
2.7	Classificação e Reconhecimento de Padrões	39
2.7.1	Máquina Vetor de Suporte	40
2.8	Medidas Visuais para Detecção de Desatenção	42
2.9	Avaliação de Desempenho	43
2.9.1	Avaliadores do Aprendizado de Máquina	43
2.9.2	Avaliadores do Detector de olhos	43
3	Metodologia	45
3.1	Aquisição do Quadro	45
3.2	Normalização do Espaço de Cor para Segmentação da Pele	47
3.3	Detecção das Formas Circulares	49
3.3.1	Detecção de Formas Circulares em toda Região da Face	50
3.3.1.1	Eliminação de Candidatos	51
3.3.2	Atualização da Posição dos Olhos Encontrados no Quadro Anterior	53
3.4	Extração de Características	56

3.5	Classificação e Validação	58
3.6	Avaliação do Nível de Desatenção do Motorista	58
4	Resultados	59
4.1	Classificador MVS	59
4.2	Detector de Olhos	60
5	Conclusão	65

Lista de Figuras

2.1	(a) Histograma da imagem alvo e (b) histograma da imagem de interesse. (c) Resultado após a especificação do histograma (Maulin Gandhi e Bhat-tacharyya, 2004).	19
2.2	(a) Imagem original com regiões de borda e (b) os elementos de bordas detectados (Sonka et al., 2008).	21
2.3	Representação da direção da borda e direção do gradiente (Sonka et al., 2008).	22
2.4	(a) Componente para aproximação das mudanças horizontais e (b) componente para aproximação das mudanças verticais (Sonka et al., 2008)	23
2.5	Modelo elíptico dos tons de pele. (Hsu et al., 2002)	30
2.6	Funcionamento básico da transformada de Hough	32
2.7	(a) Comportamento do gradiente no contorno do círculo x e y (b) posicionamento do centro em relação ao contorno na circunferência (Davies, 2004).	33
2.8	(a) Componente x e (b) componente y do núcleo \vec{O} no domínio $D(x,y)$. . .	36
2.9	(a) Imagem original e (b) após aplicação do filtro de Haar até o nível 1. . .	38
2.10	(a) Construção padrão e (b) não-padrão do filtro de Haar (Stollnitz et al., 1995).	39
2.11	Vetores de Suporte (sobre a linha pontilhada) (Bisognin, 2007).	42
3.1	Etapas da metodologia proposta.	45
3.2	Câmera TV-301SH.	46
3.3	Pinnacle Studio 710.	46
3.4	Disposição da câmera no interior do veículo.	46
3.5	Quadro capturado.	47

3.6	Componentes CbCr da imagem após normalização da iluminação, representados como imagem de tons de cinza de 8 bits.	48
3.7	Componentes CbCr após a transformação não linear do espaço YCbCr. . .	48
3.8	(a) Segmentação da pele sem transformação no espaço de cor e (b) com transformação no espaço de cor.	49
3.9	Face extraída através da segmentação de pele	49
3.10	Imagem original convertida para tons de cinza.	50
3.11	Após Suavização Gaussiana com núcleo 3x3.	50
3.12	Os círculos brancos são desenhados onde são encontrados candidatos a íris. . .	51
3.13	Comparação de regiões manualmente extraídas dos olhos esquerdo e direito. . .	52
3.14	Extração das regiões de 64x64 <i>pixels</i>	53
3.15	Regiões utilizadas para o cálculo de simetria em um dos candidatos encontrados através da transformada de Hough.	54
3.16	Imagem original do olho, convertida para tons de cinza	55
3.17	(a) Componente X e (b) componente Y do filtro de Sobel aplicado na Imagem.	55
3.18	(a) Componente X do Annulus Orientado aplicado ao resultado da aplicação do componente X do Sobel e (b) Componente Y do Annulus Orientado aplicado ao resultado da aplicação do componente Y do Sobel. . .	55
3.19	Renderização do acumulador da Transformada de Hough para tons de cinza de 8 bits.	56
3.20	Rastreamento dos olhos encontrados no quadro anterior.	56
3.21	Reposicionamento da região dos olhos esquerdo e direito de acordo com o movimento relativo entre eles.	57
3.22	Normalização de Variações na Iluminação	57
3.23	(a) Região de 64 × 64 <i>pixels</i> original e (b) após o 3º nível de decomposição utilizando transformada de Haar.	58

4.1	Exemplo de regiões utilizadas no treinamento. A linha superior mostra regiões contendo não-olhos e a inferior as que contêm olhos.	59
4.2	Resultado do algoritmo utilizado de detecção de olhos.	63
4.3	Resultado da detecção de piscada e cálculo das métricas de desatenção. .	64

Lista de Tabelas

4.1	Resultado do classificador de olhos	60
4.2	Resultado do rastreamento de olhos	60
4.3	Tempo de processamento das principais etapas do sistema	61

1 Introdução

Motoristas que dormem na direção causam entre 27% e 32% dos acidentes de trânsito e são responsáveis por 17% a 19% das mortes nas vias do Brasil (DETRAN-RS, 2008). A maior parte destes acidentes é resultado de alguns segundos de sono. Muitas vezes, o motorista não percebe o adormecimento e, por esta razão, não é possível definir o real motivo da colisão ou da capotagem.

Além das fatalidades, os acidentes por causa do sono também são responsáveis por um grande número de feridos e diversos prejuízos aos cofres públicos, devido aos gastos com operações de resgate, de tratamento aos feridos e reparos ao patrimônio. Sem contar com gastos a patrimônios de particulares, como carro, casas ou muros.

Uma forma de ajudar na prevenção de acidentes causados por motoristas com sono é a utilização de sistemas automáticos inteligentes capazes de identificar irregularidades no comportamento do motorista, que indiquem comprometimento da atenção e possam avisá-los da sua situação.

O objetivo deste trabalho é o desenvolvimento de uma metodologia computacional capaz de reconhecer o estado de sonolência ou fadiga do motorista através da análise dos olhos, para ser possível alertá-lo, evitando assim possíveis acidentes fatais.

Além disso, a difusão da tecnologia deste trabalho pode contribuir para uma redução significativa de acidentes, trazendo como consequência a diminuição dos custos diretos gerados pela hospitalização e procedimentos cirúrgicos desnecessários e a economia de custos indiretos devido ao afastamento das atividades profissionais de motoristas de automóveis.

A configuração da metodologia seria composta por um microcomputador de dimensões reduzidas, que poderia ser instalado no porta-luvas ou debaixo do banco do motorista, responsável pelo processamento e no mínimo uma câmera para fazer a captura das imagens, que seria instalada no painel do veículo, de forma a não atrapalhar o conforto do motorista. Para tentar contornar algumas das dificuldades de um sistema deste porte, que conta apenas com as informações visuais e com variáveis como a iluminação, diferentes posições da cabeça do motorista, vários tipos de pele, cabelo e olhos, serão utilizadas

técnicas de processamento de imagens, visão computacional e aprendizado de máquina.

1.1 Trabalhos Relacionados

Alguns trabalhos na literatura propõem métodos em tempo real para monitorar o comportamento do usuário através da região dos olhos.

No trabalho Savas (2005) foi utilizado o algoritmo CAMSHIFT e análise de componentes principais (PCA) para segmentação do rosto e para detecção das possíveis regiões dos olhos e então a utilização de contornos ativos (snakes) para extrair a região da íris e identificar o limite entre ela e a esclera. O algoritmo CAMSHIFT utiliza apenas a informação de *Hue* do espaço de cor HSV (Hue Saturation Value), o que o torna um pouco dependente da iluminação. A oclusão dos olhos por óculos, impossibilitam a localização confiável da íris. A taxa de detecção de olhos nos testes feitos foi de 85% .

A utilização de funções de projeção é feita em Feng e Yuen (1998), Bao et al. (2007) e Zhou e Geng (2004) e para identificação da face e região dos olhos. As funções de projeção tem como objetivo quantificar as linhas e colunas da imagem para identificar quais apresentam mais *pixels* de valores claros ou escuros. Nos dois primeiros trabalhos as funções de projeção são utilizadas para encontrar os olhos na imagem, pois tanto na posição vertical quanto na horizontal existem muitos *pixels* de pouca intensidade, devido a cor da íris, que geralmente é mais escura comparada com outras regiões do rosto. No terceiro trabalho a função de projeção é utilizada para detectar o nível de atenção do usuário, pois também é capaz de diferenciar entre olhos abertos e fechados.

Em Su et al. (2006) é proposto um sistema que identifica a face através de segmentação utilizando a informação de cor da pele. Para identificar a região dos olhos é verificada a parte superior do rosto encontrado e posteriormente aplicado o algoritmo *p-tile* para identificar a primeira *p* porcentagem de *pixels* escuros e o algoritmo *k-means* com $k = 3$. As duas regiões superiores representam os olhos e a orientação da face é estimada através de análise do triângulo formado pelos pontos que representam os dois olhos e o centro da face. São disparados sinais caso a orientação da face não indique atenção para a estrada. São feitos 3 testes que simulam a direção em um carro. A avaliação é feita sobre o tempo com os olhos fechados e o tempo de distração facial, a taxa média de acerto foi de 92.5%.

Em D´Orazio et al. (2007), são utilizadas técnicas que se aproveitam da característica circular da íris combinando com redes neurais. A metodologia possui um bom desempenho em tempo real porque o círculo é uma curva de fácil manipulação e olhos são capazes de fornecer características que os diferenciam do resto do rosto. A taxa de acerto utilizando seis pessoas diferentes para classificação da região dos olhos foi de 98.4% em testes feitos em laboratório e 95% para testes feitos dentro de um veículo.

No trabalho de Jiao e He (2007) é apresentada uma metodologia para detecção da íris em ambientes escuros utilizando o infravermelho. Nessas situações a íris se destaca das demais partes do rosto, pois ela apresenta um brilho característico (efeito *bright pupil*). A inovação do trabalho consiste no desenvolvimento de um algoritmo para a detecção dessas regiões: *Round Template Two Values Matching* (RTTVM). São feitas comparações de tempo de processamento e taxa de acerto com a análise do histograma e Transformada de Hough. Os resultados apresentados sugerem a utilização da metodologia em tempo real e com taxa de acerto de 96,4% .

O trabalho de Luhandjula (2007) utiliza um esquema baseado na detecção de face e a partir dessa informação, calcula a posição dos olhos ou a sua ausência. Os algoritmos para segmentação da face utilizados são: utilização da propriedade de simetria, uma vez que o rosto apresenta simetria com relação ao seu centro e detector de faces Viola-Jones utilizando características extraídas pelo filtro de haar e AdaBoost como classificador. Os olhos são encontrados após eliminação das outras regiões do rosto (como boca e nariz) e classificação das regiões também utilizando Adaboost. Os teste foram feitos utilizando 200 imagens estáticas e a taxa de acerto total foi 84%.

No trabalho de Fan et al. (2008) são utilizadas tanto as informações dos olhos quanto as informações da boca (detecção de bocejo), para verificar o nível de sono do motorista. Para a detecção de olhos e piscadas são utilizados AdaBoost e um operador de extração de textura LBP (Local Binary Pattern). A detecção da boca e análise de seus movimentos é feito utilizando *wavelets* Gabor e uma técnica de análise discriminante, LDA (Linear Discriminant Analysis). Os resultados apresentados não são sobre vídeos, mas sobre quadros isoladas de diferentes vídeos, com diversos tipos de pessoas, iluminação e poses. Do total de 4200 imagens a taxa de acerto para detecção de olhos fechados foi de 99.38% e de detecção de bocejo de 91.97%

O trabalho de D´Orazio et al. (2007) foi o que teve melhor desempenho dos

analisados, por apresentar bons resultados e por ser promissor em uma aplicação em tempo real. Neste trabalho foi proposto um novo algoritmo de Transformada de Hough e a representação através de transformadas de Wavelets das áreas de interesse para treino e verificação do classificador. Por esse motivo foi utilizado como base para o sistema desenvolvido nesta dissertação.

As contribuições feitas com relação aos trabalhos citados envolvem a utilização de mecanismos de normalização de iluminação para o contorno de alguns problemas encontrados em situações reais devido a variação da iluminação e a utilização de mecanismos que permitem corrigir as distorções de orientação da região de interesse (olhos) para aumentar o desempenho de classificação da metodologia, retirando a necessidade de treinamento de diversas orientações de olhos.

1.2 Organização do trabalho

Além da introdução o trabalho está dividido nos seguintes capítulos:

- No Capítulo 2, explica-se de forma resumida as teorias que fundamentam as técnicas utilizadas no trabalho desenvolvido.
- No Capítulo 3, a metodologia utilizada é descrita em etapas, desde o passo inicial da captura dos quadros da fonte de captura de vídeo até a avaliação do nível de desatenção do motorista
- No Capítulo 4, os resultados do sistema desenvolvido na metodologia são apresentados e analisados.
- No Capítulo 5, é feita a avaliação dos resultados e as considerações finais sobre o trabalho.

2 Fundamentação Teórica

Nesse capítulo é feita uma breve descrição da teoria e das técnicas utilizadas na metodologia.

2.1 Pré-processamento

As operações de pré-processamento são aquelas que atuam diretamente nas imagens após serem capturadas, quando as únicas informações que podem ser extraídas delas são os valores de intensidade da matriz que as representam. Devido a diversos tipos de situações e sensores, a imagem capturada pode conter muita informação não necessária ou apresentar falhas.

O objetivo do pré-processamento é permitir que as imagens capturadas se aproximem o máximo possível daquilo que a imagem deveria representar, em situações ideais, para os próximos passos de análise.

As técnicas de pré-processamento são divididas de acordo com o tipo de região de *pixels* utilizada, que pode ser tão pequena quanto um *pixel* ou tão grande quanto toda a imagem (Sonka et al., 2008). Os mais básicos, que consideram apenas a informação de um único *pixel* são subdivididos em dois tipos: transformações nem níveis de cinza e transformações utilizando posição do *pixel*.

2.1.1 Transformações em Níveis de Cinza

Essas operações de ponto apenas substituem o valor do *pixel* por um novo valor dado através de um mapeamento que depende da aplicação. Geralmente não é possível voltar para imagem original após essas operações. O objetivo principal é aumentar o contraste da imagem para permitir melhor uma visualização (Sonka et al., 2008).

2.1.1.1 Especificação do Histograma

O histograma de uma imagem permite a extração de informações importantes para utilização de operações para realce do contraste da imagem. Existem técnicas automáticas para correção do contraste, como a equalização do histograma, mas que apenas são capazes de aproximar o resultado para uma imagem com distribuição uniforme. Em algumas situações é mais útil especificar o formato do histograma que a imagem processada deve se aproximar. Essa técnica é chamada de especificação do histograma (Gonzalez e Wintz, 2002).

O processo de especificação do histograma é feito pela conexão do histograma alvo e do histograma da imagem de interesse, através das transformações para definição de uma distribuição uniforme obtidas da equalização do histograma .

A formulação discreta da transformação de equalização do histograma é dada por:

$$s_k = T(r_k) = (L - 1) \sum_{j=0}^k p_r(r_j) \quad (2.1)$$

Onde T é uma transformação produz um nível s_k para todo nível r_k . p_r é a função de densidade probabilidade da imagem de interesse. L é a quantidade de níveis de cinza.

De forma similar

$$G(z_q) = (L - 1) \sum_{i=0}^q p_z(z_i) \quad (2.2)$$

Onde p_z é a função de densidade probabilidade do histograma alvo. Para um valor de q tal que

$$G(z_q) = s_k \quad (2.3)$$

onde $p_z(z_i)$ é o i -ésimo valor do histograma especificado. O valor desejado z_q é então dado por

$$z_q = G^{-1}(s_k) \quad (2.4)$$

Levando em consideração que os valores de intensidade de imagens digitais são discretos, não é necessário calcular a função inversa e a especificação do histograma é feita utilizando os seguintes passos (Gonzalez e Wintz, 2002):

1. Calcular o histograma p_r da imagem de entrada e fazer a equalização do histograma utilizando a Equação 2.1;

2. Computar os valores da função de transformação G usando a Equação 2.2 para todos os valores de q , que para imagens de 8 bits se estende na faixa de 0 a 255. Armazenar os valores em uma tabela;
3. Para todos os valores de s_k , utilizar os valores de G calculados na etapa 2 para encontrar o valor que mapeia s em z . Isso é feito buscando a posição de G que possua o valor mais próximo possível de s_k . O índice do valor encontrado indica o valor de mapeamento;
4. Produzir a imagem com o histograma especificado através da ligação entre todo valor de pixel equalizado s_k da imagem de entrada no valor do histograma alvo equalizado z_q correspondente, através do mapeamento encontrado no passo 3.

A Figura 2.1 ilustra os histogramas da imagem de entrada, da imagem alvo e do histograma resultante após o processo de especificação.

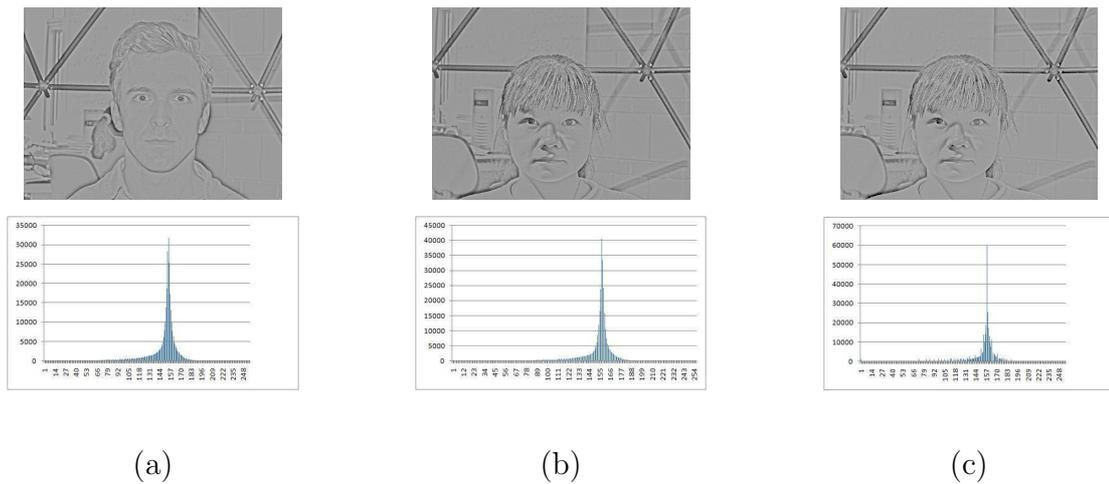


Figura 2.1: (a) Histograma da imagem alvo e (b) histograma da imagem de interesse. (c) Resultado após a especificação do histograma (Maulin Gandhi e Bhattacharyya, 2004).

2.1.2 Transformações Utilizando Posição do *pixel*

São operações que além do nível de cor de um determinado ponto levam em consideração também a sua posição. Na metodologia proposta nessa dissertação foi utilizado o Erro Médio Absoluto nessa categoria, para aproveitar a simetria entre os olhos direito e esquerdo para diminuir o custo computacional na busca da íris.

2.1.2.1 Erro Médio Absoluto (EMA)

O Erro Médio Absoluto (EMA) é uma medida simples de similaridade que é calculada através da comparação individual de *pixels* correspondentes entre duas regiões. No cálculo do EMA será levado em consideração que as regiões são espelhadas verticalmente. A equação que define o EMA é

$$\text{EMA} = \frac{\sum_{i=1}^N \sum_{j=1}^M |a_{ij} - b_{i(M-j+1)}|}{NM} \quad (2.5)$$

onde M e N representam, respectivamente, o número de colunas e de linhas das duas regiões comparadas. a_{ij} e $b_{i(M-j+1)}$ são os *pixels* das regiões que estão sendo comparadas.

Quanto mais semelhantes as regiões são, mais o valor do cálculo do EMA entre as duas se aproxima de zero.

2.2 Pré-Processamento Local

Neste grupo de operações, que recebem o nome de filtros, os novos valores dos *pixels* da imagem, têm relação com os valores dos de sua vizinhança. De acordo com aplicação estes podem ser divididos em duas categorias: suavizadores e operadores de gradiente (Sonka et al., 2008).

2.2.1 Suavizadores

Reduzem os efeitos dos ruídos nas imagens. A operação é baseada em atribuir um novo valor a um *pixel* de acordo com uma média dos valores dos *pixels* de sua vizinhança. Devido a natureza dos cálculos a imagem resultante tem uma aparência borrada e por esse motivo, apresenta perdas nas regiões que contêm informação de borda. Na metodologia proposta foi utilizado a suavização gaussiana.

2.2.1.1 Suavização Gaussiana

A função gaussiana satisfaz um grande número de propriedades matemáticas e descreve uma vasta variedade de fenômenos físicos e probabilísticos. Uma propriedade importante para a suavização é que a função gaussiana é simétrica, ela enfatiza os *pixels*

mais próximos em relação aos mais distantes. Essa propriedade reduz o borramento enquanto ainda mantém propriedades médias de ruído.

A função gaussiana em duas dimensões é dada por:

$$g(i, j) = ce^{-\frac{i^2+j^2}{2\sigma^2}} \quad (2.6)$$

onde c é constante de normalização, e é o número de Euler e σ^2 a variância.

2.2.2 Operadores de Gradiente

Os operadores de gradiente tem a função de indicar as bordas, que são regiões da imagem onde existem mudanças abruptas na sua função de intensidade (Sonka et al., 2008). Tais mudanças são importantes tanto para o processamento de imagem como para interpretação e reconhecimento de formas na visão computacional. Em alguns casos, apenas as bordas da imagem são suficientes para o seu entendimento e dos objetos que a compõem. Por esse motivo, operações que estimam a borda da imagem geralmente são utilizadas para diminuir a carga de processamento nos passos seguintes.

A Figura 2.2 ilustra os elementos de uma imagem natural e como eles são representados após aplicação de um operador de gradiente.

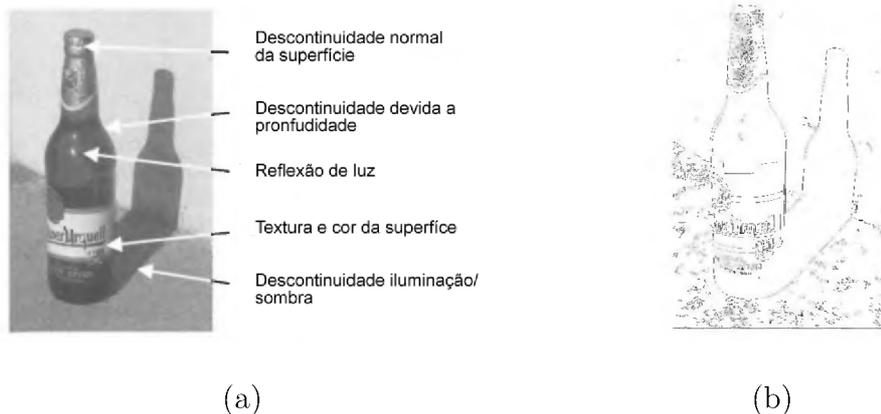


Figura 2.2: (a) Imagem original com regiões de borda e (b) os elementos de bordas detectados (Sonka et al., 2008).

O rótulo de borda é uma propriedade atribuída a alguns *pixels* e é calculado levando em conta também o comportamento de seus vizinhos. É uma variável vetor (Sonka et al., 2008) com informações de magnitude, que fornecem o magnitude do gradiente, e

direção, representando o valor naquele ponto e a direção do maior aumento na função gradiente, conforme pode ser observado na Figura 2.3.

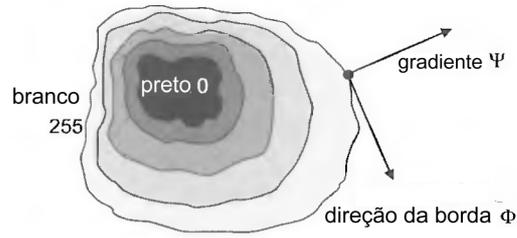


Figura 2.3: Representação da direção da borda e direção do gradiente (Sonka et al., 2008).

A mudanças bruscas da intensidade podem ser localizadas através das derivadas da imagem, pois no Cálculo, mudanças em funções contínuas podem ser descritas com derivadas (Sonka et al., 2008), e no caso bidimensional das imagens através de derivadas parciais. Então a magnitude M e direção ψ são representadas por

$$M = \sqrt{\left(\frac{\partial g}{\partial x}\right)^2 + \left(\frac{\partial g}{\partial y}\right)^2} \quad (2.7)$$

$$\psi = \arg\left(\frac{\partial g}{\partial x}, \frac{\partial g}{\partial y}\right) \quad (2.8)$$

onde ψ é um ângulo em radianos do eixo x ao ponto (x,y) .

Em imagens discretas, um cálculo aproximado da derivada parcial pode ser feito utilizando a definição de derivada (Tomasi, 2007).

$$\frac{\partial F(x, y)}{\partial x} = \lim_{\Delta x \rightarrow 0} \frac{F(x + \Delta x, y) - F(x, y)}{\Delta x} \quad (2.9)$$

Com imagens digitais o menor valor que Δx pode assumir é 1, então

$$\frac{\partial F(x, y)}{\partial x} \Big|_{x=j, y=i} \approx f(i, j + 1) - f(i, j) \quad (2.10)$$

A Equação 2.10 é apenas uma diferença entre valores discretos de dois *pixels* consecutivos no mesmo eixo e pode ser feita através de convolução com uma máscara g , com $g[0] = 1$ e $g[1] = -1$. Essa aproximação, que leva em consideração apenas uma direção, geralmente é melhorada de acordo com o detector, para tentar contornar algumas dificuldades: como bordas que não são bem detectadas em todas as direções ou erros estimando a direção da borda.

2.2.2.1 Detector de Bordas Sobel

O operador utiliza dois núcleos 3x3 que são convolucionados na imagem original para calcular a aproximação das derivadas, sendo uma para as mudanças horizontais, Figura 2.4(a) e uma para as mudanças verticais, Figura 2.4(b) .

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

(a) (b)

Figura 2.4: (a) Componente para aproximação das mudanças horizontais e (b) componente para aproximação das mudanças verticais (Sonka et al., 2008)

2.3 Transformações Geométricas

As transformações geométricas são as operações complementares às operações de ponto. São feitas principalmente para encontrar a relação de duas imagens através da eliminação de distorções geométricas que podem ter sido feitas no processo ou no momento da captura. Para identificar a distorção, é necessário encontrar o mapeamento entre os *pixels* da imagem de entrada, que é a imagem capturada, e os da imagem de saída, que deve ser semelhante à imagem que queremos fazer a relação. Esse mapeamento pode ser direto (partindo dos *pixels* de entrada para os da saída) ou inverso (dos *pixels* da saída para os da entrada) (Sonka et al., 2008).

2.3.1 Translação

A translação no plano é simplesmente o movimento do objeto. Isto é feito modificando suas coordenadas em x e/ou em y adicionando uma quantidade de movimento T_x ao longo do eixo x ou T_y ao longo do eixo y. E com isso um ponto qualquer (x,y) do objeto original passa a ser (x',y'), dados por

$$x' = x + T_x \tag{2.11}$$

$$y' = y + T_y \quad (2.12)$$

Em forma de vetor, é representado pela soma dos vetores da posição atual e do vetor deslocamento.

$$\begin{bmatrix} x' & y' \end{bmatrix}^T = \begin{bmatrix} T_x & T_y \end{bmatrix} + \begin{bmatrix} x & y \end{bmatrix}^T \quad (2.13)$$

2.3.2 Escala

Mudança de tamanho da imagem original através do aumento ou diminuição de pontos da imagem original. Isto é feito multiplicando cada ponto do objeto por um fator de escala, que pode ser no eixo x (S_x) ou no eixo y (S_y)

$$x' = x \times S_x \quad (2.14)$$

$$y' = y \times S_y \quad (2.15)$$

que matricialmente é representada por

$$\begin{bmatrix} x' & y' \end{bmatrix}^T = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} x & y \end{bmatrix}^T \quad (2.16)$$

2.3.3 Rotação

A rotação consiste em girar um objeto segundo um ângulo θ dado com relação à origem. Como inicialmente o ponto pode ser representado através de uma rotação ϕ em relação ao centro, o novo ponto será dado por

$$x' = r \cdot \cos(\theta + \phi) = r \cdot \cos\phi \cdot \cos\theta - r \cdot \sin\phi \cdot \sin\theta \quad (2.17)$$

$$y' = r \cdot \sin(\theta + \phi) = r \cdot \sin\phi \cdot \cos\theta + r \cdot \cos\phi \cdot \sin\theta \quad (2.18)$$

ou utilizando a coordenada dos pontos

$$x' = x \cos(\theta) - y \sin(\theta) \quad (2.19)$$

$$y' = y \cos(\theta) + x \sin(\theta) \quad (2.20)$$

ou em termos de representação em forma de matrizes:

$$\begin{bmatrix} x' & y' \end{bmatrix}^T = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x & y \end{bmatrix}^T \quad (2.21)$$

2.3.4 Espelhamento

O espelhamento consiste em substituir a posição dos pontos do objeto original pela posição que seria obtida caso um espelho fosse posicionado no eixo escolhido para o espelhamento e o novo objeto representasse o objeto antigo refletido. As equações que definem o espelhamento são:

$$x' = x \times Rf_x \quad (2.22)$$

$$y' = y \times Rf_y \quad (2.23)$$

ou matricialmente

$$\begin{bmatrix} x' & y' \end{bmatrix}^T = \begin{bmatrix} Rf_x & 0 \\ 0 & Rf_y \end{bmatrix} \begin{bmatrix} x & y \end{bmatrix}^T \quad (2.24)$$

A Transformação de reflexão pode ser vista como um caso especial de escala, quando o valor assume -1.

As transformações estão todas definidas na origem. Então para se obter o efeito desejado da transformação (ou do conjunto de transformações), o ponto deve sofrer uma translação que a posicione para a origem.

Todas as transformações apresentadas, exceto a translação, podem ser concatenadas, multiplicando as matrizes antes de aplicá-las aos pontos, e multiplicando apenas a matriz resultante. Com isso o número de operações necessárias é reduzida, o que computacionalmente torna o processo mais rápido. É importante lembrar também que a ordem de aplicação de um conjunto de transformações é importante, uma vez que a propriedade de multiplicação em matrizes não é comutativa.

2.3.5 Coordenadas homogêneas

O problema da representação da translação, que impossibilita o seu uso no processo de concatenação, pode ser resolvido com o uso de coordenadas homogêneas. No sistema de coordenadas homogêneas é adicionado uma nova coordenada ao ponto. Então ao invés de representá-lo por um par de números (x, y) , ele é representado pela tripla (x, y, W) . Um ponto pode possuir diferentes representações em coordenadas homogêneas, desde que elas sejam múltiplas de uma delas. Ao menos uma das coordenadas precisa ser não-zero. Quando a coordenada W é não-zero, é possível dividir as coordenadas por seu

valor, o que representa a nova coordenada $(x/W, y/W, 1)$ e as novas coordenadas x e y são chamadas de coordenadas cartesianas do ponto homogêneo (Foley, 1996).

Devido a nova representação dos pontos, que passaram a possuir uma nova coordenada, as matrizes precisam aumentar suas dimensões para permitir a multiplicação com os pontos para aplicar as transformações. A nova representação da matriz de translação é dada por

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix}^T = \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x & y & 1 \end{bmatrix}^T \quad (2.25)$$

As matrizes das outras transformações geométricas em coordenadas homogêneas são dadas pela adição de uma nova linha e uma nova coluna após as já existentes com valores 0 em todas as posições com exceção da representada pela última linha e coluna, que recebe o valor 1.

Como exemplo, a nova transformação de rotação definida em coordenadas homogêneas.

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix}^T = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x & y & 1 \end{bmatrix}^T \quad (2.26)$$

2.4 Variação de Iluminação

Sistemas que utilizam imagens como entrada são cada vez mais explorados em diversas aplicações como comércio, segurança, autenticação e outros. Infelizmente esses sistemas na maioria das vezes são dependentes do dispositivo de captura e sobretudo da iluminação.

Muitos algoritmos falham na sua aplicação direta na imagem de entrada pois a variação da iluminação no momento da captura pode ser muito grande. Em aplicações de reconhecimento de faces, por exemplo, a variação de iluminação de uma única face pode gerar mais diferenças que comparação entre indivíduos (Adini et al., 1997). Então para que sistemas de processamento de imagens possam ter a capacidade de fornecer respostas adequadas em diversos tipos de ambiente, é necessário atenuar os efeitos da variação de iluminação.

Na metodologia desenvolvida foram utilizadas técnicas classificadas nas categorias principais de correção de iluminação (Vuçini et al., 2007): extração de características invariantes e formas canônicas. Além destas também foi utilizado a normalização de cor que utiliza o conceito de branco referencial (Li, 2002).

2.4.1 Extração de Características Invariantes à Iluminação

Extração de características que não sofrem tanto impacto quando ocorre mudança de iluminação. Alguns exemplos são: as derivadas da intensidade da imagem, convolução com filtros 2D de Gabor, imagem quociente.

Na metodologia desenvolvida foi utilizado as informações extraídas das derivadas parciais de intensidade da imagem aproximadas utilizando os núcleos do operador Sobel.

2.4.2 Formas Canônica

Como tentativa de normalizar a imagem, são utilizadas as formas canônicas. As variações são descartadas com uma transformação que gera uma forma canônica e apenas então a imagem é submetida às outras etapas do sistema.

2.4.2.1 Single Scale Retinex

As diferenças entre a qualidade da cena e de sua imagem capturada devem-se grande parte ao fato de que a faixa dinâmica de valores de intensidade da imagem ser muito maior que o do dispositivo de captura. Existem técnicas que buscam reduzir essa diferença através do mapeamento de uma faixa dinâmica muito grande para uma muito menor. Tais técnicas são classificadas na categoria de compressão de faixa dinâmica. Um outro fator para baixa qualidade de captura é a dependência da cor gravada com relação à variação de iluminação.

Técnicas de constância de cor têm como objetivo tornar as cores similares, mesmo quando a cena é submetida a variações de intensidade e variação de iluminação. O Single Scale Retinex (SSR) (Maulin Gandhi e Bhattacharyya, 2004) proporciona alto nível de compressão de faixa dinâmica e constância de cor.

O SSR é definido pela equação:

$$R_i(x, y) = \ln I_i(x, y) - \ln [F(x, y) \otimes I_i(x, y)] \quad (2.27)$$

onde $I_i(x, y)$ é o componente da imagem na banda i , e $i \in (R, G, B)$, o termo $[F(x, y) \otimes I_i(x, y)]$ representa a imagem convoluída com um filtro Gaussiano.

2.4.3 Normalização da Cor

O processo de normalização da cor tem como objetivo aproximar as respostas das técnicas aplicadas na imagem mesmo com variações de iluminação.

A normalização utilizada é baseada no conceito de cores utilizadas como referência de branco. Arbitrariamente os 5% maiores valores de intensidade de iluminação encontrados na imagem são considerados como branco (branco de referência) se a quantidade de *pixels* com esse valor é maior que um limiar definido. Caso isso aconteça os componentes RGB (Red Green Blue) de todos os *pixels* da imagem são linearmente modificados de forma que o valor médio das cores consideradas como branco é escalado para 255 (para imagens com 8 bits em cada um dos componentes) (Li, 2002).

2.5 Segmentação

Em aplicações reais, as imagens em uma visão macro, geralmente são constituídas de diversos elementos e cada um deles possuem características que permitem diferenciar as imagens umas das outras. As diferenciações podem ser simples, como separar um objeto do fundo da cena ou podem ser mais complicadas como diferenciar entre diversos tipos de objetos.

Em processamento de imagens, a segmentação consiste em separar a imagem em grupos, onde cada grupo é formado por *pixels* que contém uma ou várias relações que os permitem ficar juntos, e também que os separem dos componentes de outros grupos. Consiste no primeiro passo para a localização e representação dos objetos da cena.

Por essa razão, a segmentação é considerada, dentre todas as etapas do processamento de imagens, a etapa mais crítica do tratamento da informação. É nesta etapa que são definidas as regiões de interesse para processamento e análise posteriores. Como

consequência deste fato, quaisquer erros ou distorções presentes nesta etapa se refletem nas demais etapas, de forma a produzir ao final do processo resultados não desejados que possam contribuir de forma negativa para a eficiência de todo o processamento (Gonzalez e Wintz, 2002).

O tipo de segmentação utilizada depende da aplicação, e mesmo dentro da própria aplicação a segmentação pode necessitar de vários ajustes para uma separação mais precisa dos grupos. Por isso além das várias operações de segmentação existentes, atualmente existe um grande interesse no desenvolvimento de outras.

O metodologia desenvolvida durante o período da dissertação utiliza as segmentações baseadas no tom da pele através o modelo YCbCr e a Transformada de Hough como segmentação baseada em bordas (Sonka et al., 2008) .

2.5.1 Segmentação Baseada na Cor da Pele

O modelo utilizado para identificação da pele é dependente do espaço de cor e da região dentro desse espaço que limita os tons definidos como pele. O espaço YCbCr é um dos mais utilizados para esse tipo de modelagem pois é perceptualmente uniforme, permite uma grande separação entre luminância e crominância e uma boa compatidade das regiões de pele (Hsu et al., 2002).

Para aumentar ainda mais a independência entre luminância e crominância do modelo YCbCr é necessário aplicar uma transformação não linear (Hsu et al., 2002) para melhorar a robustez da identificação de regiões de pele. A transformação do croma é definida por

$$C'_i(Y) = \begin{cases} (C_i(Y) - \bar{C}_i(Y)) \cdot \frac{W_{c_i}}{W_{c_i}(Y)} + \bar{C}_i(K_h) & \text{se } Y < K_l \text{ ou } K_h < Y, \\ C_i(Y) & \text{se } Y \in [K_l, K_h] \end{cases} \quad (2.28)$$

onde $W_{c_b}(Y)$ e $W_{c_r}(Y)$ representam a extensão do modelo definido por

$$W_{c_i}(Y) = \begin{cases} WL_{c_i} + \frac{(Y - Y_{min}) \cdot (W_{c_i} - WL_{c_i})}{K_l - Y_{min}} & \text{se } Y < K_l, \\ WH_{c_i} + \frac{(Y_{max} - Y) \cdot (W_{c_i} - WH_{c_i})}{Y_{max} - K_h} & \text{se } K_h < Y \end{cases} \quad (2.29)$$

Os centros do modelo de cor \bar{C}_b e \bar{C}_r são calculados através de

$$\bar{C}_b(Y) = \begin{cases} 108 + \frac{(K_l - Y) \cdot (118 - 108)}{K_l - Y_{min}} & \text{se } Y < K_l, \\ 108 + \frac{(Y - K_h) \cdot (118 - 108)}{Y_{max} - K_h} & \text{se } K_h < Y, \end{cases} \quad (2.30)$$

$$\bar{C}_r(Y) = \begin{cases} 154 - \frac{(K_l - Y) \cdot (154 - 144)}{K_l - Y_{min}} & \text{se } Y < K_l, \\ 154 + \frac{(Y - K_h) \cdot (154 - 132)}{Y_{max} - K_h} & \text{se } K_h < Y, \end{cases} \quad (2.31)$$

Após a transformação do modelo YC_bC_r o valor do tom será considerado pelo se satisfazer o modelo elíptico representado pela Figura 2.5. Os pontos vermelhos representam as tonalidades consideradas como pele, os pontos interiores da elipse são os que são considerados pele pelo algoritmo de segmentação e os pontos azuis são outros tipos de tonalidade, que não representam pele.

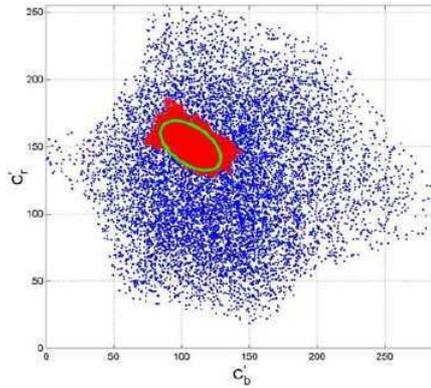


Figura 2.5: Modelo elíptico dos tons de pele. (Hsu et al., 2002)

A equação do modelo é dada por

$$\frac{(x - ec_x)^2}{a^2} + \frac{(y - ec_y)^2}{b^2} = 1 \quad (2.32)$$

x e y são encontrados através da equação

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} C'_b - c_x \\ C'_r - c_y \end{bmatrix} \quad (2.33)$$

2.5.2 Segmentação Baseada em bordas

Esse tipo de segmentação é utilizada após a localização das bordas da imagem. Porque como explicado na Seção 2.2.2, em algumas aplicações, apenas as informações de borda são relevantes para o reconhecimento do objeto. Mas apenas a detecção de borda não é suficiente para identificar um possível objeto, é preciso segmentar as bordas de forma que cada cadeia de bordas separadas em um grupo represente um objeto (Sonka et al., 2008).

Aproveitando a propriedade de circularidade da íris, a segmentação baseada em bordas utilizada neste trabalho foi a transformada de Hough.

2.5.2.1 Transformada de Hough

O funcionamento básico da transformada de Hough, que é compartilhado pela extensão da Transformada de Hough em todas as curvas e formas que ela pode ser utilizada para fazer o reconhecimento, é a utilização de um espaço paramétrico. Em formas simples como retas, circunferências e elipses, o espaço paramétrico está relacionado com a representação paramétrica de cada forma.

Uma curva ou reta representa um ponto no seu espaço paramétrico que não é nada mais que um espaço acumulador onde, em processamento de imagens digitais, cada posição representa o incremento da posição respectiva de um *pixel* de borda da imagem original aplicada na equação paramétrica da forma de interesse. Quanto maior o números de *pixels* que serão levados ao mesmo ponto no espaço acumulador pela equação paramétrica, maior a probabilidade desse ponto representar a forma procurada.

Em formas mais complexas, as abordagens mais utilizadas são dividí-las em formas simples e utilizar a transformada de Hough em cada uma das formas, buscando uma coerência espacial nas formas finais, ou aplicando a Transformada Generalizada de Hough, quando possível.

Transformada de Hough na Circunferência

A utilização da Transformada de Hough para detectar círculos é talvez a aplicação mais direta da técnica e mais fácil de visualizar, embora sejam necessárias modificações no método original para melhorar a acurácia e a velocidade e permitir a detecção de círculos com raios variáveis.

O primeiro método que utilizou a Transformada de Hough para círculos em imagens digitais era composto dos seguintes passos:

1. Estimativa do gradiente de intensidade em todas as posições da imagem.
2. Limiarização do gradiente para identificar as possíveis bordas.
3. Construção de uma imagem que representa o espaço paramétrico (acumulador), congruente com o espaço da imagem.
4. Para todos os pontos da borda, incrementar todas as posições a distância R (raio, conhecido *a priori*) de cada um no espaço acumulador.

5. Buscar o espaço acumulador por picos, que correspondem aos centros prováveis de círculos na imagem.

Na Figura 2.6 é possível observar o funcionamento básico da técnica. O círculo na imagem original de bordas é representado na Figura 2.6 (a). É representado de forma pontilhada pois o número de pontos é reduzido para facilitar a visualização, o primeiro incremento no espaço acumulador é feito de uma região circular em volta de um dos pontos da borda, Figura 2.6 (b). Após a repetição do processo para ou outros pontos da borda, é possível verificar que a posição que apresenta o maior número de incrementos representa o centro do círculo, como observado na Figura 2.6 (c).

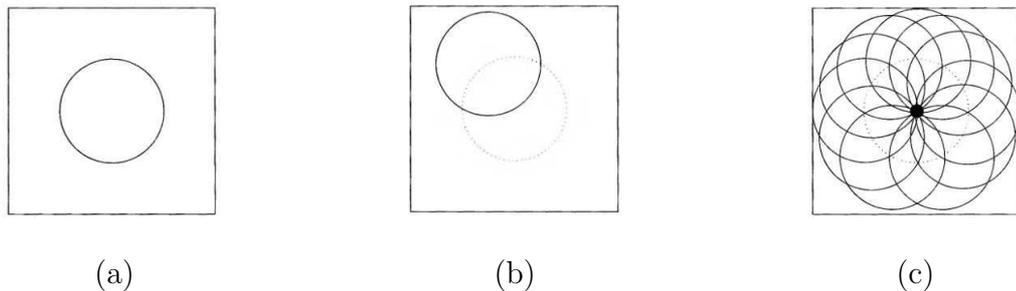


Figura 2.6: Funcionamento básico da transformada de Hough

Abordagem de Gerig e Klein

A abordagem de Gerig e Klein (Liatsis e Nazarboland, 2002) (Yuen et al., 1990) (Davies, 2004) para a localização de círculos é uma outra forma de implementação baseado na Transforma de Hough original que possui algumas melhorias.

- Utilização de três acumuladores com dois parâmetros;
- Projeção reversa do espaço paramétrico;
- Adicionalmente é possível utilizar informação de direção da borda.

São utilizados três parâmetros acumuladores de mesmo tamanho da imagem: C_w , acumulador semelhante ao da definição original; C_r , armazena o raio dos valores altos encontrados nos acumuladores; C_p , armazena o tamanho do acumulador nas posições de valores altos.

Esse tipo de transformada não permite encontrar círculos concêntricos, por outro lado é possível encontrar o mais completo.

Projeção Reversa do Espaço Paramétrico

Imperfeições nos objetos podem gerar regiões com valores altos no espaço paramétrico ao invés de pontos, o que pode dificultar muito a identificação de vários círculos.

Na projeção reversa um novo espaço paramétrico é construído. A transformada de Hough é novamente aplicada, mas dessa vez apenas a posição de maior valor no acumulador para cada pixel de borda, encontrada anteriormente é incrementado (Davies, 2004).

Dessa forma objetos muito fragmentados ou oclusos têm maiores chances de serem encontrados e torna o desempenho da transformada de Hough menos dependente da escolha de um limiar na interpretação do espaço acumulador.

Utilização de informação da borda

Em círculos a direção do gradiente em cada ponto aponta para o centro ou para direção inversa ao centro. Aproveitando essa propriedade é possível reduzir o número de posições que serão incrementadas no acumulador. A Figura 2.7 ilustra essa propriedade.

Idealmente apenas um ponto precisaria ser incrementado, para cada raio da faixa de busca. Mas na prática é incrementado um arco pois muitas vezes existem distorções na imagem capturada.

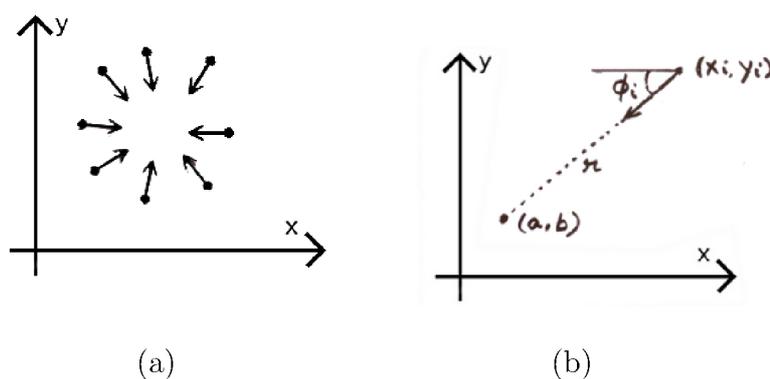


Figura 2.7: (a) Comportamento do gradiente no contorno do círculo x e y (b) posicionamento do centro em relação ao contorno na circunferência (Davies, 2004).

Abordagem de Atherton e Kerbyson

A transformada de Hough na circunferência também pode ser formulada como

um operador de convolução (Atherton e Kerbyson, 1999), ou filtro da transformada de Hough, para detecção de círculo de tamanho único, ou detecção de círculos de tamanhos entre um limite inferior e um superior. Os operadores de convolução citados são: Círculo Único, Annulus, Annulus Vetor Orientado, Annulus Fase Codificado. Os núcleos são construídos como segue:

a) Círculo único

Detecção de Círculos com raio único

$$O_{SC}(m, n) = \begin{cases} 1, & \text{se } (R - 0.5)^2 < m^2 + n^2 < (R + 0.5)^2 \\ 0, & \text{caso contrário} \end{cases} \quad (2.34)$$

onde R é o comprimento do raio, m e n são as coordenadas horizontais e verticais de um ponto do núcleo.

b) Annulus

Detecção de Círculos de raios de tamanhos variáveis:

$$O_A(m, n) = \begin{cases} 1, & \text{se } (R)_{min}^2 < m^2 + n^2 < (R)_{max}^2; R_{max} + 1 \\ 0, & \text{caso contrário} \end{cases} \quad (2.35)$$

onde R_{min} é o menor raio e R_{max} o maior do intervalo de variação .

c) Annulus Orientado

Detecta uma variação de círculos, como o núcleo annulus, mas também utiliza informação de orientação das bordas tomando o produto escalar entre a orientação da borda e o campo de orientação dentro do annulus. O operador de convolução contém dois componentes de vetor unitário em cada posição dentro do annulus. O operador de convolução é dado por $O_{OA}(O_{OA_x}, O_{OA_y})^T$ e é definido por

$$O_{OA_x}(m, n) = \begin{cases} \cos\theta, & \text{se } (R)_{min}^2 < m^2 + n^2 < (R)_{max}^2 \\ 0, & \text{caso contrário} \end{cases} \quad (2.36)$$

$$O_{OA_y}(m, n) = \begin{cases} \sin\theta, & \text{se } (R)_{min}^2 < m^2 + n^2 < (R)_{max}^2 \\ 0, & \text{caso contrário} \end{cases} \quad (2.37)$$

onde $\theta_{mn} = \tan^{-1}(n/m)$

Aplicando o Núcleo do Annulus Orientado

O operador escolhido para detecção das regiões circulares foi o annulus orientado. A aplicação dos componentes dos operadores podem ser feitos em uma imagem de magnitude de borda ou em uma imagem com gradiente parcial em x e y .

A operação do annulus orientado requer um produto escalar entre o gradiente da imagem e o vetor de annulus orientado. A contribuição total feita em um único ponto no acumulador é a soma das contribuições individuais de cada ponto na imagem original. Expandindo o produto escalar, pode ser escrito como a soma de duas convoluções. Uma entre o gradiente da imagem em x e o componente x do annulus orientado, e o segundo entre o gradiente y e o componente y do annulus orientado (Atherton e Kerbyson, 1999).

Em D´Orazio et al. (2007) é utilizado uma transformada de Hough baseada em uma modificação do annulus orientado, que além do conceito anteriormente descrito adiciona constantes de normalização para permitir que os picos do acumulador sejam nas regiões mais circulares e não mais contrastadas. Esse novo operador é descrito da seguinte forma:

$$(x, y) = \frac{\int \int_{D(x,y)} \vec{g}(\alpha, \beta) \cdot \vec{O}(\alpha - x, \beta - y)}{2\pi(R_{min} - R_{max})} \quad (2.38)$$

onde o domínio $D(x, y)$ é definido por

$$D(x, y) = (\alpha, \beta) \in R^2, |R_{min}^2 \leq (\alpha - x)^2 + (\beta - y)^2 \leq R_{max}^2 \quad (2.39)$$

\vec{g} é o vetor gradiente definido por

$$\vec{g}(x, y) = \left[\frac{\partial I(x, y)}{\partial x}, \frac{\partial I(x, y)}{\partial y} \right]^T \quad (2.40)$$

\vec{O} é o vetor núcleo definido por

$$\vec{O}(x, y) = \left[\frac{\cos(\arctan(y/x))}{\sqrt{x^2 + y^2}}, \frac{\sin(\arctan(y/x))}{\sqrt{x^2 + y^2}} \right]^T \quad (2.41)$$

No domínio discreto essa transformada de Hough é implementado como dois filtros, aplicados nas derivadas parciais da imagem (D´Orazio et al., 2007). A Figura 2.8 ilustra cada um dos componentes.

2.6 Extração de Atributos

A extração de atributos consiste em obter informações da imagem, relevantes para aplicação. Geralmente as fases anteriores do processamento de imagem têm como



Figura 2.8: (a) Componente x e (b) componente y do núcleo \vec{O} no domínio $D(x,y)$.

objetivo corrigir os erros ou remover regiões não necessárias para que as medidas dos atributos utilizados sejam as mais precisas possíveis.

A escolha errada no número de atributos nessa fase pode acarretar uma baixa caracterização da forma e muitos erros no processo de reconhecimento. Em alguns casos pode ser interessante normalizar cada eixo para construir um espaço que seja bem distribuído e facilite o processo de classificação.

Nesse trabalho, são utilizadas transformadas de wavelets para extração de atributos que serão passados ao classificador.

2.6.1 Transformadas de Wavelets

Existem poucas situações práticas, onde os dados analisados ou as funções que descrevem um determinado processo podem ser modeladas de uma maneira simples. O ideal é poder utilizar funções que possuem um comportamento conhecido para descrever dados ou funções com que se está trabalhando. Um dos pioneiros na utilização de funções que se sobrepondo podem aproximar o comportamento de outras funções, foi Fourier (Graps, 1995). A transformada que leva o seu nome permite a expansão de uma função através da combinação linear de senos e cossenos. Embora o trabalho tenha sido pioneiro e tenha modificado a forma como as funções eram vistas na época, a utilização da transformada de Fourier possui limitações. No espaço dos sinais por exemplo, essa transformada é capaz de fornecer informações sobre todas as frequências na imagem, mas não a de onde elas estão presentes. Do estudo de funções que fossem capazes de suprir essas falhas, surgiram as funções wavelets.

Wavelets são recursos matemáticos que têm como objetivo decompor funções hierarquicamente. Permitindo que as funções sejam descritas em termos gerais, como também é capaz de representar os níveis de detalhes presentes.

Em termos gerais a decomposição de uma função $f(t)$ em um conjunto de

funções base ψ utilizando transformada de wavelet de uma dimensão no caso contínuo é dada por (Sonka et al., 2008)

$$c(s, \tau) = \int_{\mathbb{R}} f(t) \psi_{s,\tau}(t) dt, s \in \mathbb{R}^+ - \{0\}, \tau \in \mathbb{R} \quad (2.42)$$

As wavelets são generalizadas a partir de uma única wavelet mãe $\psi(t)$ através da escala s e translação τ

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t - \tau}{s}\right) \quad (2.43)$$

onde o coeficiente $\frac{1}{\sqrt{s}}$ é utilizado para normalizar a energia do sinal entre diferentes escalas

Na análise multirresolução, as transformadas de wavelets são semelhantes a um banco de filtros, o qual é constituído de um filtro L (passa-baixa) e um filtro H (passa-alta), chamados de *Quadrature Mirror Filter* (QMF). Inicialmente é preciso definir uma função de escala $\Phi(x)$ dada por (Silva et al., 2006)

$$\phi(x) = \sqrt{2} \sum_n l(n) \phi(2x - n), n \in \mathbb{Z} \quad (2.44)$$

Além da função de escala é preciso definir uma função ψ que expande as diferenças entre as resoluções, chamada de função wavelet.

$$\psi(x) = \sqrt{2} \sum_n h(n) \phi(2x - n), n \in \mathbb{Z} \quad (2.45)$$

A relação entre os dois filtros é dada por:

$$h(n) = (-1)^n * l(1 - n) \quad (2.46)$$

A transformada de Haar é a mais antiga e mais simples das transformadas de wavelets, As suas funções de escala (denotados por ψ), são dadas pelo conjunto de escalas e translações de funções *box* (Sonka et al., 2008). Os wavelets que correspondem a base *box* são chamados de Wavelets Haar e são dados por:

$$\psi_{ji}(x) = 2^{j/2} \phi(2^j x - i), i = 0, \dots, 2^j - 1 \quad (2.47)$$

Os coeficientes dos filtros de passa-baixa (L) e passa-alta (H) da transformada de Haar são dadas por (Silva et al., 2006):

$$l = \left[\begin{array}{cc} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{array} \right] \text{ e } h = \left[\begin{array}{cc} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{array} \right] \quad (2.48)$$

Um nível de decomposição da wavelet Haar é obtido aplicando os filtros QMF nas direções horizontais e verticais de uma imagem. Como pode ser visto na Figura 2.9 a decomposição da imagem forma então quatro sub-bandas: LL, que representa as frequências baixas da imagem e LH, HL e HH, que representam as frequências altas. A primeira letra representa o filtro que foi aplicado na vertical (passa-baixa (L) e passa-alta (H)) e a segunda o filtro que foi aplicado na horizontal. Os níveis de decomposição seguintes são feita apenas na banda LL.

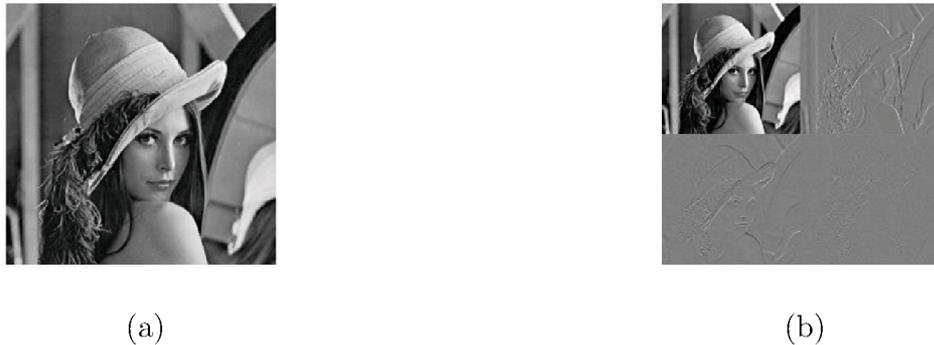


Figura 2.9: (a) Imagem original e (b) após aplicação do filtro de Haar até o nível 1.

Em imagens digitais a decomposição em níveis pode ser feita de duas formas diferentes: decomposição padrão e não padrão. A diferença entre as duas formas é que na primeira a transformada de wavelets de uma dimensão é aplicada em cada linha na imagem e depois nas colunas. Dessa forma, caso a transformada seja composta de vários níveis, o resultado é formado por aplicar todos os níveis da transformada nas linhas e então nas colunas, e os valores resultantes são todos coeficientes de detalhes exceto pelos coeficientes de média restante. Na segunda forma as operações decorrentes da transformadas de Haar (média e detalhe) são feitas recursivamente nas linhas e então nas colunas a cada nível apenas no quadrante que contém o resultado das médias nas duas direções (Stollnitz et al., 1995). Esses dois tipos são ilustrados na Figura 2.10.

Os coeficientes da transformada de Wavelet mantém informações de textura e forma do objeto da imagem. Dessa maneira é possível distinguir o objeto desejado de outros que possam ter em comum um dos dois aspectos.

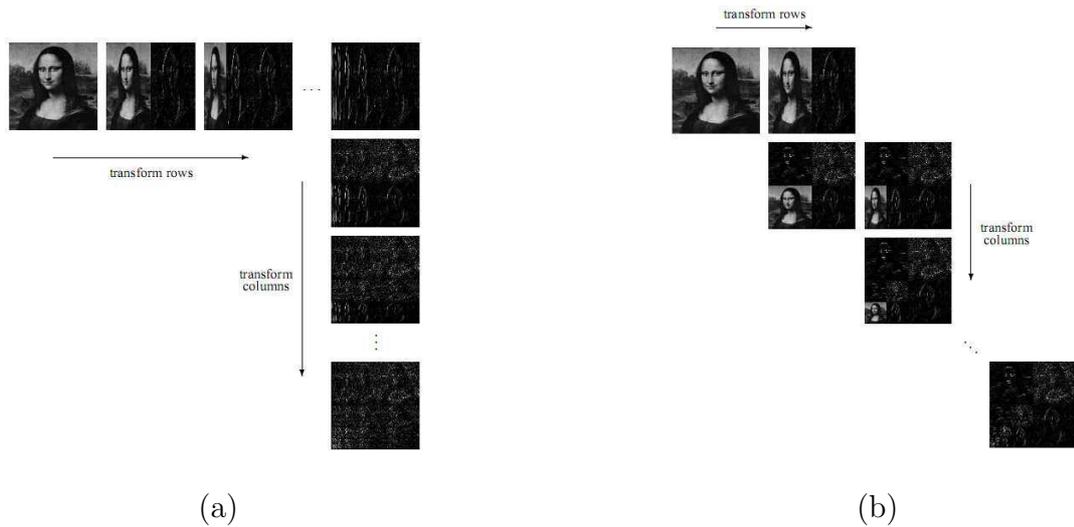


Figura 2.10: (a) Construção padrão e (b) não-padrão do filtro de Haar (Stollnitz et al., 1995).

2.7 Classificação e Reconhecimento de Padrões

Nessa etapa será feita a identificação dos objetos segmentados da imagem. Destas regiões são extraídas as informações com as operações de extração de atributos. A região passa a ser representada em um novo espaço, do qual a dimensionalidade é diretamente ligada a quantidade de informações que a fase de extração de atributos é capaz de oferecer. Na maioria dos casos o sistema precisa passar por uma fase de treinamento antes de iniciar o processo de classificação. A função dos Sistemas de Aprendizado é aproveitar informações dessa etapa para serem capazes de classificar objetos em classes distintas de acordo com a aplicação (Esquef et al., 2003).

Existem dois tipos de métodos de aprendizado: supervisionados e não supervisionados. Nos métodos supervisionados a fase de aprendizado é dependente de um agente externo que irá fornecer amostras de cada classe com as características usadas para representá-las. Apenas a partir dessas amostras o sistema é capaz de definir uma função para classificação. Nos métodos não supervisionados, não existe a necessidade de um agente externo, pois o próprio sistema de aprendizado é capaz de escolher as características para separação das amostras em diferentes classes.

Após o processo de aprendizado, objetos desconhecidos serão fornecidos ao classificador na fase de reconhecimento. No caso em que a classificação não é supervisionada, o classificador receberá os objetos desconhecidos e, a partir da medida dos diferentes

parâmetros (atributos dos objetos presentes na imagem), ele tentará alocá-los em diferentes classes. A identificação de classes é usualmente realizada a partir da identificação de agrupamentos *clusters* de objetos no espaço de medidas.

Nesse trabalho foi utilizado a Máquina de Vetor de Suporte, para identificar uma região como olho ou não-olho.

2.7.1 Máquina Vetor de Suporte

A Máquina de Vetor de Suporte (MVS) é uma técnica de aprendizagem supervisionada, usada para estimar uma função com o objetivo de classificar dados em duas classes através da construção de um hiperplano, que atua como superfície de decisão de forma que as distâncias entre as classes e o hiperplano sejam máximas. Esse hiperplano é obtido na fase de treinamento da MVS através de um conjunto de amostras composto por duas classes separáveis.

Uma das grandes vantagens da MVS é que enquanto a maioria das técnicas utilizadas para o aprendizado de máquina são elaboradas para minimizar o risco empírico, que é aumentar o desempenho apenas sobre o conjunto de treinamento, a MVS é elaborada para minimizar o risco estrutural, que é a probabilidade de classificar de forma incorreta padrões ainda não vistos por uma distribuição de probabilidade dos dados fixos e desconhecidos (Cristianini e Sawe-Taylor, 2000).

Um classificador MVS é capaz de encontrar um hiperplano baseado em um conjunto de pontos denominados vetores de suporte, o qual maximiza a margem de separação entre as classes. Por hiperplano entende-se uma superfície de separação de duas regiões em um espaço multidimensional, onde o número de dimensões possíveis pode ser, até infinito. Mesmo quando as duas classes não são separáveis, a MVS é capaz de encontrar um hiperplano através do uso de conceitos pertencentes à teoria da otimização. Seja o conjunto de amostras de treinamento (x_i, y_i) , sendo x_i no vetor de entrada y_i a classificação correta das amostras e $i = 1, \dots, n$ o índice de cada ponto amostral. O objetivo da classificação é estimar a função $f : R^n \rightarrow \{\pm 1\}$, que separe corretamente os exemplos de teste em classes distintas. A etapa de treinamento estima a função, procurando por valores de w e b tais que a seguinte relação seja satisfeita

$$y_i ((wx_i) + b) \geq 1 \quad (2.49)$$

Os valores ótimos de w e b serão encontrados ao minimizar a seguinte equação, de acordo com a restrição dada pela Equação 2.49 (Chaves, 2006).

$$\phi(w) = \frac{w^2}{2} \quad (2.50)$$

A MVS ainda possibilita encontrar um hiperplano que minimize a ocorrência de erros de classificação nos casos em que, uma perfeita separação entre as duas classes não for possível. Isso graças a inclusão de variáveis de folga, que permitem que as restrições presentes na Equação 2.49 sejam quebradas.

O problema de otimização passa a ser então a minimização da Equação 2.50, de acordo com a restrição imposta pela Equação 2.49. C é um parâmetro de treinamento que estabelece um equilíbrio entre a complexidade do modelo e o erro de treinamento e deve ser selecionado pelo usuário.

$$\Phi(w, \xi) = \frac{(w^2)}{2} + C \sum_{i=1}^N \xi_i \quad (2.51)$$

$$y_i((wx_i) + b) + \xi \geq 1 \quad (2.52)$$

Através da teoria dos multiplicadores de Lagrange, chega-se à Equação 2.53. O objetivo então passa a ser encontrar os multiplicadores de Lagrange α_i ótimos que satisfaçam a Equação 2.54 (Chaves, 2006).

$$w(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{j=1}^N \alpha_i \alpha_j \gamma_i \gamma_j (x_i x_j) \quad (2.53)$$

$$\sum_{i=1}^N \alpha_i \gamma_i = 0, 0 \leq \alpha_i \leq C \quad (2.54)$$

Apenas os pontos onde a restrição da Equação 2.49 seja exatamente igual à unidade têm correspondentes $\alpha_i \neq 0$. Esses pontos são chamados de vetores de suporte, pois se localizam geometricamente sobre as margens. Tais pontos têm fundamental importância na definição do hiperplano ótimo, pois os mesmos delimitam a margem do conjunto de treinamento.

A Figura 2.11 destaca os pontos que representam os vetores de suporte. Os pontos além da margem não influenciam decisivamente na determinação do hiperplano, enquanto que os vetores de suporte, por terem pesos não nulos, são decisivos.

Para que a MVS possa classificar amostras que não são linearmente separáveis, é necessária uma transformação não-linear que altere o espaço entrada (dados) para um

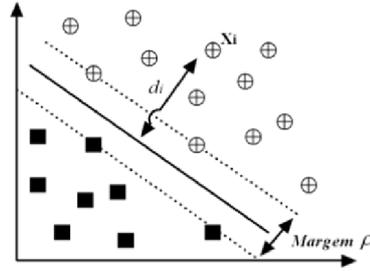


Figura 2.11: Vetores de Suporte (sobre a linha pontilhada) (Bisognin, 2007).

novo espaço (espaço de características). Esse espaço deve apresentar dimensão suficientemente grande, e através dele, a amostra pode ser linearmente separável. Dessa maneira, o hiperplano de separação é definido como uma função linear de vetores retirados do espaço de características ao invés do espaço de entrada original. Essa construção depende do cálculo de uma função K de núcleo de um produto interno (Haykin e Engel, 2001). A função K pode realizar o mapeamento das amostras para um espaço de dimensão muito elevada sem aumentar a complexidade dos cálculos. A Equação 2.55 mostra o resultado da Equação 2.53 com a utilização de um núcleo.

$$w(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{j=1}^N \alpha_i \alpha_j \gamma_i \gamma_j K(x_i x_j) \quad (2.55)$$

Uma família importante de funções de núcleo é a função de base radial, muito utilizada em problemas de reconhecimento de padrões e também utilizada neste trabalho. A função de base radial é definida por

$$K(x_i, y_i) = \exp(-\gamma \|x_i - x_j\|) \quad (2.56)$$

2.8 Medidas Visuais para Detecção de Desatenção

O nível de desatenção do motorista é baseado na medida PERCLOS (*Percentage Eyelid Closure*) pois ela pode ser considerada confiável e válida para determinar o nível de desatenção do motorista (Dinges e Grace, 1998). O cálculo aproximado do PERCLOS é feito utilizando duas medidas que são fatores para sua determinação: duração de fechamento dos olhos e a frequência de fechamento dos olhos.

Os valores utilizados para essas medidas serão descritos na Seção 3.6

2.9 Avaliação de Desempenho

Nessa seção são explicadas os avaliadores utilizados para avaliar os resultados e entender os valores encontrados com os algoritmos de aprendizado de máquina e do detector de olhos.

2.9.1 Avaliadores do Aprendizado de Máquina

Neste trabalho serão utilizados os avaliadores (Almeida, 2007): a Sensibilidade (s), Especificidade (e) e a Acurácia (a), por serem métodos estatísticos comumente utilizados. Essas medidas são utilizadas para verificar os resultados da classificação das regiões candidatas a olhos e não-olhos.

A sensibilidade é a medida da capacidade do método de decisão de prever a ocorrência de olhos quando a amostra realmente representa olho e é dado pela equação:

$$s = \frac{VP}{VP + FN} \quad (2.57)$$

onde, VP é o número de verdadeiros positivos, que são pontos definidos na imagem como olhos e classificados como olhos e FN o número de falso negativo, que são as amostras que representam olhos e foram classificados como não-olhos.

A especificidade (e) é a medida da capacidade do método de decisão de apontar ausência da condição para aqueles casos que realmente não a têm, conforme apresentado:

$$e = \frac{VN}{VN + FP} \quad (2.58)$$

onde VN é o número de verdadeiros negativos, correta classificação de regiões que apresentam não olhos. FP é o número de amostras de não-olhos, classificadas como olhos.

A acurácia (a) é a probabilidade de predições corretas, e é dada pela Equação abaixo:

$$a = \frac{VP + VN}{VP + VN + FP + FN} \quad (2.59)$$

2.9.2 Avaliadores do Detector de olhos

A avaliação do detector de olhos é dada pela taxa de acerto dada pela equação

$$Acertos = \frac{O_{enc}}{T_q - Q_{of}} \quad (2.60)$$

onde O_{enc} , representa a quantidade de quadros com olhos corretamente encontrados, T_q , o número total de quadros e Q_{of} o número de quadros nos quais os olhos estavam fechados.

O avaliador do detector de olhos indica a taxa de acerto da quantidade de quadros onde os olhos aparecem e são corretamente encontrados pela metodologia.

3 Metodologia

Nessa seção serão descritas brevemente as etapas utilizadas para o desenvolvimento da metodologia do sistema.

A metodologia proposta neste trabalho é composta de 8 etapas: captura do quadro, normalização do espaço de cor para segmentação da pele, extração da região do rosto, detecção de formas circulares, aplicação do SSR em conjunto com especificação do histograma nos candidatos a olho para eliminação baseada na simetria, extração de característica nos candidatos restantes, classificação dos candidatos restantes e avaliação do nível de atenção do usuário. O esquema da metodologia é descrito na Figura 3.1.

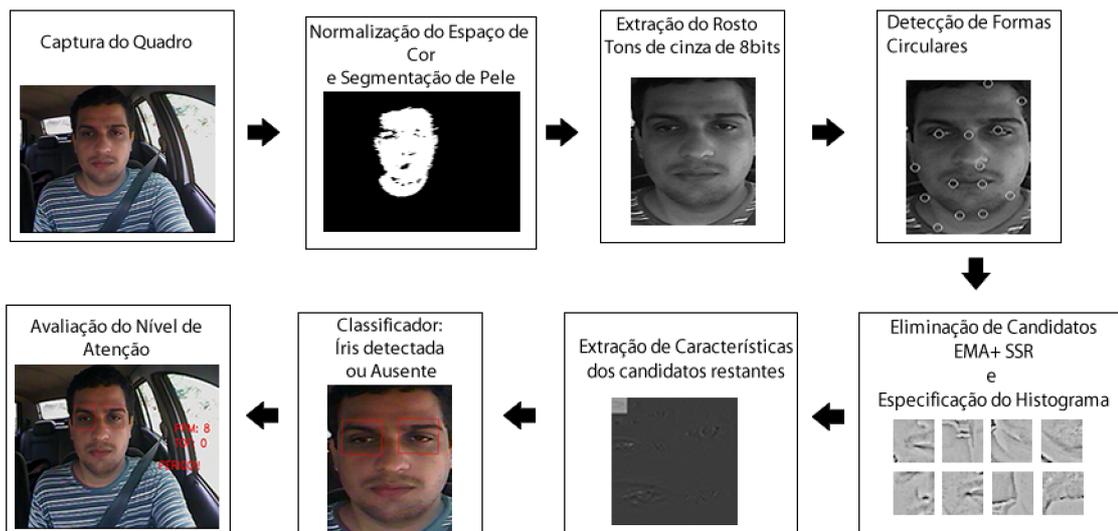


Figura 3.1: Etapas da metodologia proposta.

3.1 Aquisição do Quadro

A primeira etapa é a captura dos quadros que compõem o vídeo. A fonte do vídeo foi a mini câmera TV-301SH (Figura 3.2). Esse modelo foi escolhido devido ao seu tamanho reduzido e as seguintes características técnicas relevantes: sensor 1/3" sharp color ccd, balanço automático de branco, resolução em *pixels* de (H) 768 x (V) 494 e operações com iluminação mínima de 0.1 lux para filmagens em cor.

Por se tratar de um equipamento analógico de captura a comunicação com



Figura 3.2: Câmera TV-301SH.

computador utilizado para o processamento deve feita através de uma placa de captura de vídeo. O modelo escolhido para essa finalidade foi o Pinnacle Studio 710, Figura 3.3. As características relevantes da placa de captura são resolução em *pixels* de (H) 720 x (V) 480, gravação a 30 quadros por segundo.



Figura 3.3: Pinnacle Studio 710.

Para captura e processamento dos videos foi utilizado um *laptop* com processador de frequência 1.5 Ghz e 3Gb de memória RAM.

A disposição da câmera no interior do veículo é apresentada na Figura 3.4. A câmera é fixada ao para-brisa do carro com adesivo de silicone que permite boa aderência e não danifica o vidro no momento da retirada. Idealmente a fixação pode ser substituída por um suporte instalado no painel e permitir a correção de acordo com a relação de altura entre painel e banco do motorista, dependendo do modelo do veículo.



Figura 3.4: Disposição da câmera no interior do veículo.

A captura e manipulação do vídeo no computador é feita utilizando a Biblioteca OpenCV (Intel, 2000).

3.2 Normalização do Espaço de Cor para Segmentação da Pele

Após a captura do quadro da fonte de vídeo (Figura 3.5) é feita a segmentação da pele para identificar a provável região do rosto utilizada a fim de reduzir o espaço de busca dos olhos e reduzir o custo computacional de uma busca global.



Figura 3.5: Quadro capturado.

Para segmentar a pele com o algoritmo descrito na Seção 2.5.1, é necessário antes aplicar a normalização de iluminação baseada em branco de referência, Seção 2.4.3, pois a aparência da cor da pele é dependente da iluminação. Na imagem resultante é aplicado a transformação não linear dos componentes de cromaticidade no espaço YCbCr. Os *pixels* que estão dentro do modelo elíptico são marcados como pele.

A conversão entre o espaço de cores RGB, extraídos pela fonte de captura, para o espaço YCbCr é feita utilizando as equações (Intel, 2000)

$$Y = I_C(x, y)_R \times 0,299 + I_C(x, y)_G \times 0,587 + I_C(x, y)_B \times 0,114 \quad (3.1)$$

$$Cb = (I_C(x, y)_B - Y) \times 0,564 + 128 \quad (3.2)$$

$$Cr = (I_C(x, y)_R - Y) \times 0,713 + 128 \quad (3.3)$$

A Figura 3.6 representa os componentes de cromaticidade do espaço YCbCr da imagem de entrada representados como imagens de tons de cinza de 8 *bits*, o componente

Y não é exibido pois não participa da transformação. Na Figura 3.7 estão os resultados da aplicação da transformação do não-linear do espaço relacionado com o componente de luminância.

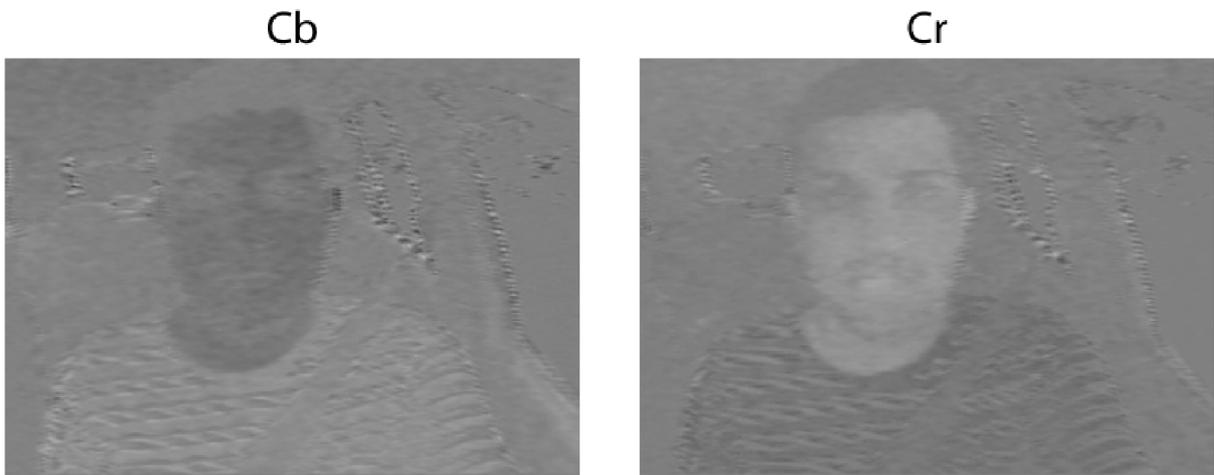


Figura 3.6: Componentes CbCr da imagem após normalização da iluminação, representados como imagem de tons de cinza de 8 bits.

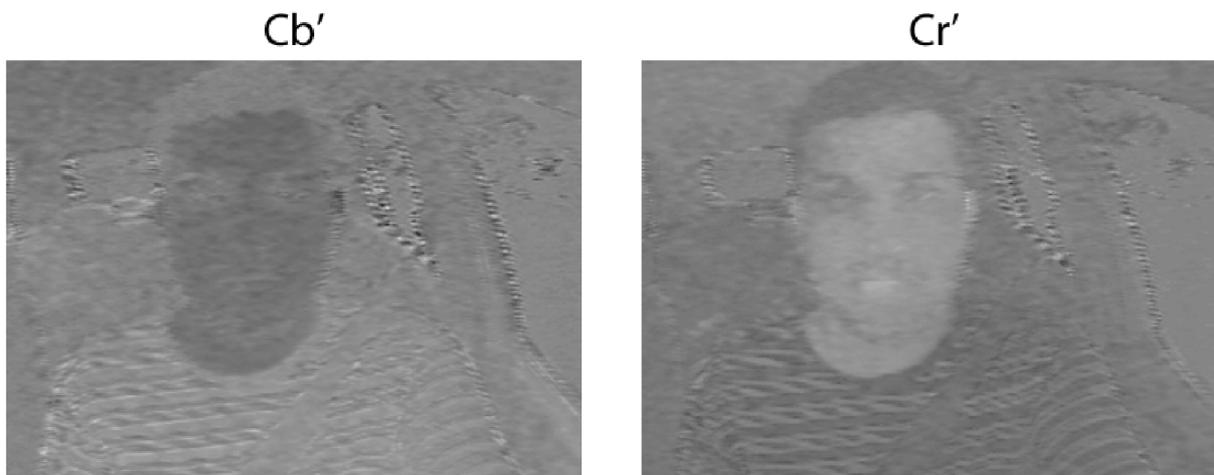


Figura 3.7: Componentes CbCr após a transformação não linear do espaço YCbCr.

A vantagem da transformação não-linear no espaço de cor pode ser observado na Figura 3.8. Após a segmentação de pele, a maior região é extraída e representa o candidato a face, Figura 3.9, todo o processamento posterior será feito apenas nessa região.

Como a informação de cor não é mais utilizada nos próximos passos do sistema, para efeito de simplicidade de manipulação, cada quadro é convertido em tons de cinza de 8, Figura 3.10 *bits*, utilizando a equação (Intel, 2000):



Figura 3.8: (a) Segmentação da pele sem transformação no espaço de cor e (b) com transformação no espaço de cor.



Figura 3.9: Face extraída através da segmentação de pele

$$I_G(x, y) = I_C(x, y)_R * 0,299 + I_C(x, y)_G * 0,587 + I_C(x, y)_B * 0,114 \quad (3.4)$$

onde $I_G(x, y)$ representa a imagem em níveis de cinza, $I_C(x, y)$ a imagem representada no espaço RGB.

Após essa etapa é utilizado uma suavização gaussiana (Seção 2.2.1) para amenizar o efeito do ruído que é comum em imagens naturais. O resultado é exibido na Figura 3.11.

3.3 Detecção das Formas Circulares

A escolha do algoritmo de Hough na circunferência para encontrar as regiões candidatas é baseada nas propriedades físicas dos olhos. A primeira delas é a mais evidente, a circularidade da íris. Essa circularidade pode ser muito aproveitada para con-



Figura 3.10: Imagem original convertida para tons de cinza.



Figura 3.11: Após Suavização Gaussiana com núcleo 3x3.

tornar o problema da rotação na identificação de objetos, pois objetos circulares são intrinsecamente invariáveis a esse tipo de transformação. A segunda delas é o contraste entre a íris e a esclera. A íris é sempre mais escura que a esclera, não importando a sua cor. Como a transformada de Hough é aplicada nas imagens de magnitude de bordas ou no caso, suas componentes, e as bordas são regiões que apontam mudanças bruscas no gradiente da imagem, o contorno da íris irá na maioria dos casos aparecer como borda.

São utilizadas dois tipos de transformada de Hough para detectar regiões com formas circulares. A primeira delas, com o menor tempo de processamento, é aplicada em toda a imagem resultante da etapa de detecção da face e é denominada transformada de Hough de Gerig and Klein (Liatsis e Nazarboland, 2002; Davies, 2004), quando não se sabe anteriormente a posição dos olhos. A segunda, é baseada em convolução (Atherton e Kerbyson, 1999; D´Orazio et al., 2007) e é aplicada para atualizar a posição da região dos olhos no quadro anterior.

3.3.1 Detecção de Formas Circulares em toda Região da Face

A detecção de formas circulares em toda região da face, ilustrado na Figura 3.12, é feita utilizando a abordagem da transformada de Hough feita por Gerig e Klein (Yuen et al., 1990).



Figura 3.12: Os círculos brancos são desenhados onde são encontrados candidatos a íris.

3.3.1.1 Eliminação de Candidatos

Nas etapas seguintes, é necessário extrair mais informações dos candidatos para conseguir obter uma separação ainda mais precisa entre olhos e não-olhos. Para isso são retiradas da imagem em tons de cinza de 8 *bits* regiões de 64x64 *pixels* centradas na posição que representa um pico no acumulador da transformada de Hough.

A relação entre essa região $I_{(x',y')}$ de 64x64 *pixels* e a imagem completa $I_{(x,y)}$, em tons de cinza, é feita utilizando transformações geométricas. As transformações utilizadas são: translação, rotação e espelhamento

$$\text{Translação} = \begin{bmatrix} 1 & 0 & I_{(x',y')} - \frac{64}{2} \\ 0 & 1 & I_{(x',y')} - \frac{64}{2} \\ 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

$$\text{Rotação} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

$$\text{Espelhamento} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.7)$$

onde a posição do -1 da matriz indica espelhamento em torno do eixo y.

O espelhamento é feito arbitrariamente apenas para o olho direito, ou para posição suposta do olho direito. O objetivo é ter apenas um posicionamento de olho, para

facilitar os próximos passos. Na Figura 3.13(a) a orientação do olho direito da forma é como extraída da imagem. Na Figura 3.13(b) o resultado da aplicação do espelhamento no olho direito em torno do eixo y e considerando o seu centro na origem.

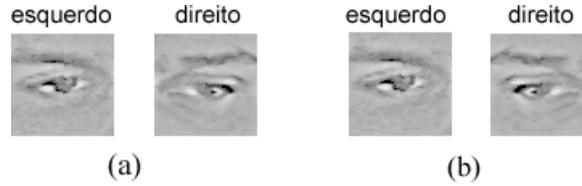


Figura 3.13: Comparação de regiões manualmente extraídas dos olhos esquerdo e direito.

$$T_{result} = T_{rot} \times T_{trans} \times T_{esp} \quad (3.8)$$

onde T_{result} representa a matriz de transformação resultante, T_{rot} a matriz de rotação, T_{trans} a de translação e T_{esp} a de espelhamento.

$$I_{(x,y)} = T_{result} \times I_{(x',y')} \quad (3.9)$$

onde $I_{(x,y)}$ representa a posição na imagem original e $I_{(x',y')}$ a posição na imagem de 64×64 pixels.

Após a definição da matriz de transformação, é possível extrair a região da imagem original. O posicionamento inicial da região de 64×64 pixels é ilustrado na Figura 3.14(a). A Figura 3.14(b) mostra a posição após a aplicação da matriz de transformação geométrica. Na Figura 3.14(c) é feita a localização da região na imagem de tons de cinza de 8 bits e na Figura 3.14(d) a sub-imagem extraída. O ângulo considerado para matriz de rotação é de -20° e é utilizado apenas como exemplo.

É possível eliminar candidatos baseando-se na propriedade física de similaridade dos olhos. Se a região é realmente um olho, então existe uma região simétrica a ela que também representa olho, mas de forma espelhada. A medida EMA (Seção 2.1.2) é utilizada para verificar numericamente esse grau de similaridade. Os valores do EMA para identificar a semelhança foram obtidos empiricamente como menores que 18, uma vez que embora a semelhança total forneça o valor 0, por diferenças causadas por iluminação ou mesmo física os valores são maiores que esse.

Como não se sabe inicialmente se a região representa o olho direito ou esquerdo e a inclinação da cabeça, a busca da região simétrica utilizadas o cálculo do EMA é

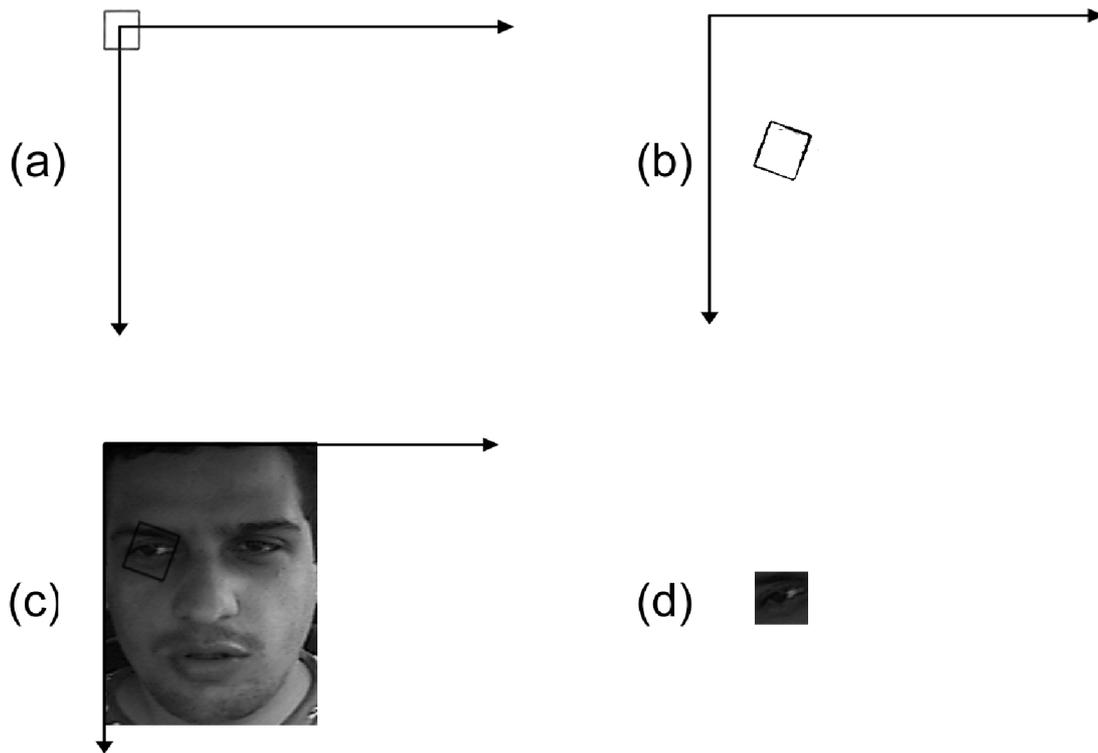


Figura 3.14: Extração das regiões de 64×64 *pixels*

feito no sentido direito e esquerdo e em vários ângulos de rotação como mostra a Figura 3.15. As extrações são feitas apenas na região do rosto na imagem em tons de cinza. As sobreposições dos retângulos são feitas nas imagens coloridas apenas para facilitar a visualização. O quadrado azul representa a região de 64×64 *pixels* extraída com centro em um dos acumuladores da transformada de Hough, o quadrado de outra cor e com mesma orientação é a região verificada por simetria.

A vantagem dessa etapa é que ela atualiza a matriz da transformação geométrica com informação de rotação. Com isso é possível corrigir a inclinação da cabeça e apresentar apenas um tipo de orientação para o algoritmo de aprendizado de máquina MVS (que também pode ser treinado com apenas um tipo de orientação).

3.3.2 Atualização da Posição dos Olhos Encontrados no Quadro Anterior

Para essa fase foi utilizada a transformada de Hough Annulus Orientado. Trata-se de um operador com custo computacional alto por ser baseado em convolução, mas apresenta o resultado mais robusto entre todos os algoritmos analisados para o de-

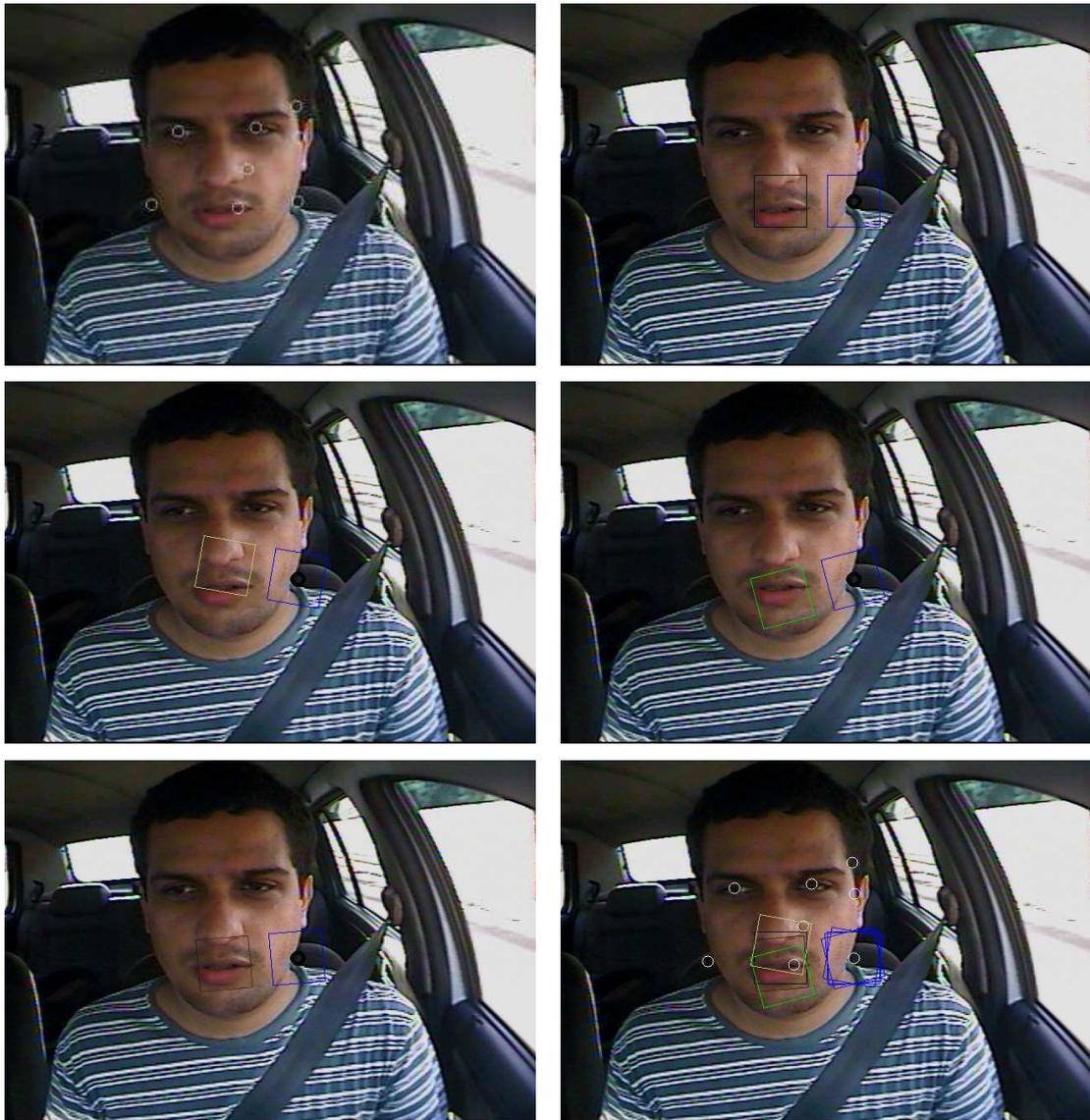


Figura 3.15: Regiões utilizadas para o cálculo de simetria em um dos candidatos encontrados através da transformada de Hough.

envolvimento do sistema.

Como exigência da aplicação do Annulus Orientado, é necessário calcular as derivadas parciais do quadro atualmente processado nas suas componentes em x e em y . Por serem capazes de oferecer uma boa representação dessas derivadas os núcleos do operador de Sobel (Seção 2.2.2) são utilizados. A convolução de cada um de seus núcleos sobre a imagem de entrada Figura 3.16 tem como saída uma imagem que representa uma derivada parcial da imagem. Então as matrizes do filtro de Sobel são aplicadas e representam a derivada em X Figura 3.17(a) e Y Figura 3.17 (b).

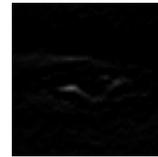
Após esse cálculo, é feita então a convolução dos componentes do núcleo da



Figura 3.16: Imagem original do olho, convertida para tons de cinza



(a)



(b)

Figura 3.17: (a) Componente X e (b) componente Y do filtro de Sobel aplicado na Imagem.

transformada de Hough (definidos na Seção 2.5.2) com os seus equivalentes em orientação na imagem: Sua componente em X é convoluída na derivada parcial em x da imagem (Figura 3.18 (a)) e sua componente em y na derivada parcial em y da imagem (Figura 3.18 (b)).



(a)



(b)

Figura 3.18: (a) Componente X do Annulus Orientado aplicado ao resultado da aplicação do componente X do Sobel e (b) Componente Y do Annulus Orientado aplicado ao resultado da aplicação do componente Y do Sobel.

Para obter o resultado esperado do produto escalar no Anulus orientado o resultado das duas convoluções é somado e a imagem resultante (Figura 3.19) representa o acumulador da transformada de Hough. Quanto mais próximo do branco, maior a chance dessa posição representar o centro de um círculo e em consequência o centro dos olhos.

A movimentação simétrica dos olhos é utilizada para diminuir o custo computacional da busca dos olhos. A ideia é posicionar no quadro atual duas janelas nas posições dos olhos detectadas no quadro anterior, e aplicando a segunda transformada de Hough nessas regiões, é possível reposicionar essas janelas de acordo com movimento dos

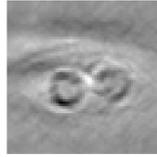


Figura 3.19: Renderização do acumulador da Transformada de Hough para tons de cinza de 8 bits.

picos dos seus acumuladores. Inicialmente a posição dos olhos encontrada no quadro anterior é aproveitada Figura 3.20 (a), então é feita a extração das regiões do olho esquerdo e direito da imagem de tons de cinza, Figura 3.20(b) e por fim a identificação do pico no acumulador da transformada de Hough (centro da íris) Figura 3.20(c).

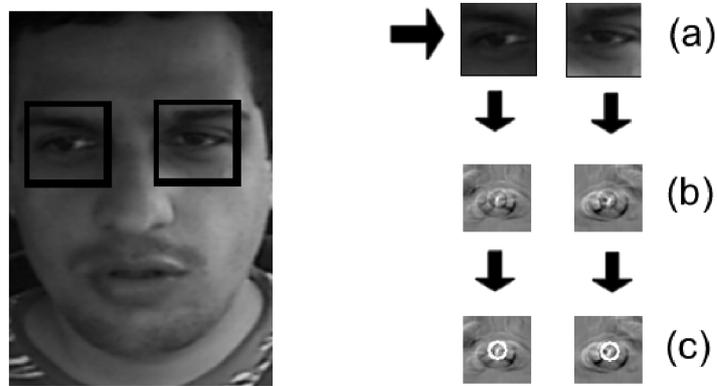


Figura 3.20: Rastreamento dos olhos encontrados no quadro anterior.

A movimentação dos dois olhos em um mesmo sentido significa uma translação, se um deles se movimenta para cima e outro para baixo isso indica uma inclinação para o lado direito ou esquerdo. A posição das janelas é recalculada de acordo com o movimento. A Figura 3.21 ilustra esse procedimento. A primeira linha representa as translações horizontais, a segunda linha representa as translações verticais e na terceira linha estão ilustrados os exemplos de rotação.

3.4 Extração de Características

Antes das outras operações nessa abordagem, o Single Scale Retinex em conjunto com a especificação do histograma é aplicado nas regiões de 64×64 *pixels* restantes, com o objetivo de contornar alguns dos efeitos da variação de iluminação (Figura 3.22). A primeira linha corresponde as imagens de entrada, a segunda linha é o resultado da aplicação do Retinex e na terceira linha o retinex após especificação do histograma.

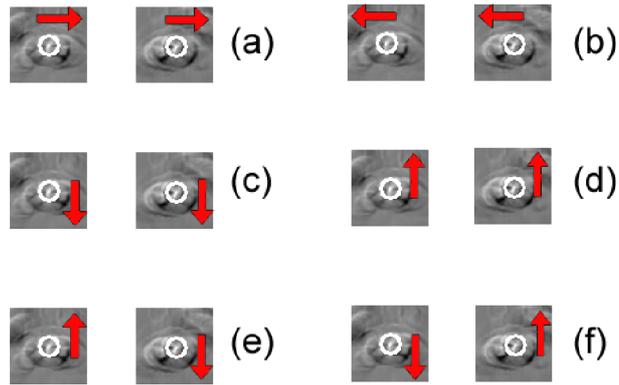


Figura 3.21: Reposicionamento da região dos olhos esquerdo e direito de acordo com o movimento relativo entre eles.

Esse procedimento também é aplicado às imagens de treinamento do classificador e tem como objetivo reduzir as diferenças causadas pela iluminação e aumentar o desempenho da classificação.



Figura 3.22: Normalização de Variações na Iluminação

Na imagem resultante do processo de amortização da variação de iluminação é aplicada a transformada de Haar até o terceiro nível, e apenas os coeficientes de terceiro nível são utilizados para representá-los no classificador. O terceiro nível (Figura 3.23(b)) foi escolhido porque o número de seus coeficientes representa apenas $\frac{1}{16}$ do total de *pixels* do quadro de 64×64 *pixels* original, mas contém informações suficientes de forma e textura para diferenciar as regiões dos olhos de outras regiões. A redução do número de características reflete-se na diminuição do tempo de processamento na fase de classificação.



Figura 3.23: (a) Região de 64×64 *pixels* original e (b) após o 3º nível de decomposição utilizando transformada de Haar.

3.5 Classificação e Validação

Para verificar as regiões que realmente representam olhos, é utilizado o algoritmo de aprendizado de máquina, o MVS (Seção 2.7.1). O primeiro passo é treiná-lo. Para isso são feitas extrações de forma manual da região dos olhos que serão treinadas como amostras positivas e de outras regiões da face ou do ambiente que serão treinadas como regiões negativas. A representação dessas regiões no MVS também utiliza os coeficientes da transformada de Haar (Seção 2.6.1) de terceiro nível de cada uma das regiões extraídas. O treinamento é feito *offline*.

3.6 Avaliação do Nível de Desatenção do Motorista

A duração do tempo dos olhos fechados (TOF) é feita conferindo quantos segundos passaram desde a última vez que os olhos foram encontrados. As piscadas por minuto (PPM) são calculadas contando quantas vezes em um intervalo de 1 minuto os olhos não são encontrados.

Para as duas medidas são definidos dois limiares. O primeiro é para alertar sobre pequenas anomalias e o segundo para quando o comportamento visual chega em um nível perigoso. Nos vídeos de testes os limiares para o TOF são 5 e 15, respectivamente, e para o PPM 4 e 10.

Em situações reais, com tempo de análise mais amplos, os valores são maiores. O objetivo da escolha desses limiares foi apenas para verificar se a metodologia é capaz de analisar o vídeo e indicar situações anormais.

4 Resultados

Para validar a metodologia proposta utilizamos a região dos olhos com dimensões de 64×64 *pixels*. Essa dimensão foi escolhida de forma empírica e depende da resolução da fonte de captura. A distância entre os olhos para o cálculo da medida de similaridade EMA fica no intervalo $[86,94]$ *pixels* e foi obtida através de testes realizados em pessoas de diferentes idades e pequenas variações de distância em relação a câmera.

Os resultados são relacionados com a aplicação da metodologia em dois vídeos. Estes vídeos possuem duas pessoas diferentes dirigindo em situação real, executando movimentos naturais e submetidos a variação de iluminação. A duração dos vídeos é entre 3:30 e 6:50 minutos. Os resultados da aplicação da metodologia nos casos de teste podem ser encontrados em <http://www.youtube.com/user/leolabpai/videos?view=u>.

4.1 Classificador MVS

O MVS foi treinado utilizando 40 amostras de imagens representando olhos e 200 representando não-olhos. A extração dessas regiões foi feita manualmente de imagens que contêm diversos tipos de localização da íris em relação ao centro do olho e expressões de rosto. A Figura 4.1 exemplifica algumas das regiões usadas para treinamento.



Figura 4.1: Exemplo de regiões utilizadas no treinamento. A linha superior mostra regiões contendo não-olhos e a inferior as que contêm olhos.

Na Tabela 4.1 é possível visualizar os resultados do classificador MVS aplicado nos vídeos de teste. Os números de VP e VN são relacionado ao tempo que os olhos estão fechados (ou não foram encontrados). Quanto maior esse tempo maior o número de regiões

VN, quanto menor esse tempo, maior o número de regiões VP.

Tabela 4.1: Resultado do classificador de olhos

	Vídeo 1	Vídeo 2
VP	17791	7303
FP	110	197
VN	3641	4844
FN	273	70
$s(\%)$	98.48	99.05
$e(\%)$	97.06	96.09
$a(\%)$	98.24	97.84

4.2 Detector de Olhos

A Tabela 4.2 mostra os resultados encontrados pelo algoritmo de rastreamento de olhos.

Tabela 4.2: Resultado do rastreamento de olhos

	Vídeo 1	Vídeo 2
Total de Quadros	10162	5243
Quadros com olhos fechados	322	215
Quadros com olhos encontrados	8926	3932
Quadros com olhos não encontrados	639	695
Taxa de acerto(%)	93.33	84.97

Os resultados em conjunto com o tipo de movimento da pessoa nos vídeos indicam que o sistema é capaz de permitir liberdade dos movimentos naturais do motorista e manter uma boa capacidade de detecção dos olhos, o que evitaria alarmes desnecessários que poderiam acabar por distrair o motorista.

A Figura 4.2 exemplifica alguns quadros de um vídeo com o rastreamento dos olhos em diversos tipos de orientação do rosto. Os quadrados vermelhos representam as regiões que foram validadas pelo SVM. A taxa de quadros por segundo foi reduzido para 10 para melhor visualização.

Na Figura 4.3 é ilustrado a detecção de piscadas e a atualização dos índices que indicam o nível de sono dos motoristas. PPM significa piscadas por minuto e é atualizado cada vez que o usuário pisca e zerado a cada minuto. O significado de TOF é o tempo com olhos fechados, o valor é atualizado enquanto o usuário permanece com os olhos fechados e zerado quando os olhos são abertos.

Na Tabela 4.3 está descrito o tempo necessário para as principais etapas da metodologia. Dependendo do conhecimento prévio da posição dos olhos nos quadro anteriores (Seção 3.3.1 ou Seção 3.3.2), algumas das etapas não são executadas.

Tabela 4.3: Tempo de processamento das principais etapas do sistema

Etapa	Tempo(s)
Validação MVS	0.0011
Transformada de Hough em todo o quadro	0.0322
Transformada de Hough apenas na região de interesse	0.0133
Transformada de Hough na região do olho	0.0020
Retinex e Especificação do Histograma nas regiões do olho	0.004
Segmentação de Pele (normalização e transformação do espaço de cor)	0.050

Para entender os valores da tabela é preciso lembrar que o sistema captura a taxa de 25 quadros por segundo. Para fornecer um desempenho em tempo real cada quadro precisa ser processado em no máximo $1/25 = 0.04$ segundos. Em média, quando as condições de iluminação são favoráveis, são validadas 4 regiões iniciais (os picos encontrados na transformada de Hough) como ponto da busca de olhos e para região do olho encontrada são validados em média 6 regiões em busca do outro olho por simetria. Então durante o período inicial de descoberta da posição dos olhos o desempenho do algoritmo é $1/(0,05 + 0,032 + 4*(0,001 + 0,004) + 6*(0,001 + 0,006))$ ou de forma aproximada, 7 quadros por segundo. Com o processamento apenas das regiões onde os olhos foram achados anteriormente esse valor sobe para 27,5 quadros por segundo, que é capaz de atender parcialmente a exigência de tempo real da aplicação.

Baseado nos testes efetuados, foi verificado que a metodologia apresenta robustez para permitir ao usuário liberdade de movimentação e ainda sim, garantir o rastreamento correto, enquanto seus olhos estiverem abertos e direcionados para frente do veículo. A não detecção dos olhos quando o motorista olha para os lados atrapalha a

identificação do nível de sonolência do motorista, mas os índices de desatenção calculados nesses casos podem ser úteis para evitar batidas causados por alguma distração que desvia o olhar do motorista para frente.

Os mecanismos utilizados para contornar os efeitos das variações de iluminação também apresentaram resultados satisfatórios nos vídeos de testes. A busca por regiões circulares, baseada nas derivadas parciais das imagens, que são características resistentes a variações de iluminação, permitiriam a identificação de círculos na íris enquanto a propriedade de contraste entre íris e esclera é mantida. O classificador, treinado apenas com imagens retiradas em um ambiente de iluminação controlada, foi capaz de diferenciar corretamente a região dos olhos de regiões contendo não olhos em situações de variação de iluminação ou iluminação parcial da face.

Entretanto, mesmo com a utilização de técnicas para controle da iluminação, alguns casos extremos falham. Algumas vezes a luz ofuscante em conjunto com propriedades no rosto do motorista como oleosidade ou algum tipo de produto tornam toda a região da face branca ou reflexiva, impossibilitando o sucesso da utilização do sistema.

Outro problema detectado durante os testes diz respeito ao próprio funcionamento do método. Como estão sendo procuradas regiões circulares o algoritmo não é capaz de verificar se os olhos estão fechados, mas apenas se ele não encontrou os olhos na cena. Essa dificuldade implica que se as informações não encontradas de um olho serão utilizadas como sinônimo de olho fechado. O algoritmo precisa ainda ser mais refinado, pois mesmo os resultados encontrados pelo detector de olhos, nos testes efetuados, ultrapassando os 89% dos que realmente aparecem no vídeo, a escala de quadros no processamento de vídeos pode ocasionar o aparecimento frequente de erros. Como por exemplo: a taxa de captura utilizada nesse trabalho foi de 25 quadros por segundo, o que significa que a cada segundo existe uma taxa média de erro em 2,5 quadros(11%). Caso os erros apareçam seguidos, que é o que geralmente ocorre, em intervalos de tempo razoáveis o sistema pode indicar um comportamento perigoso do motorista e gerar sinais de alerta desnecessários.

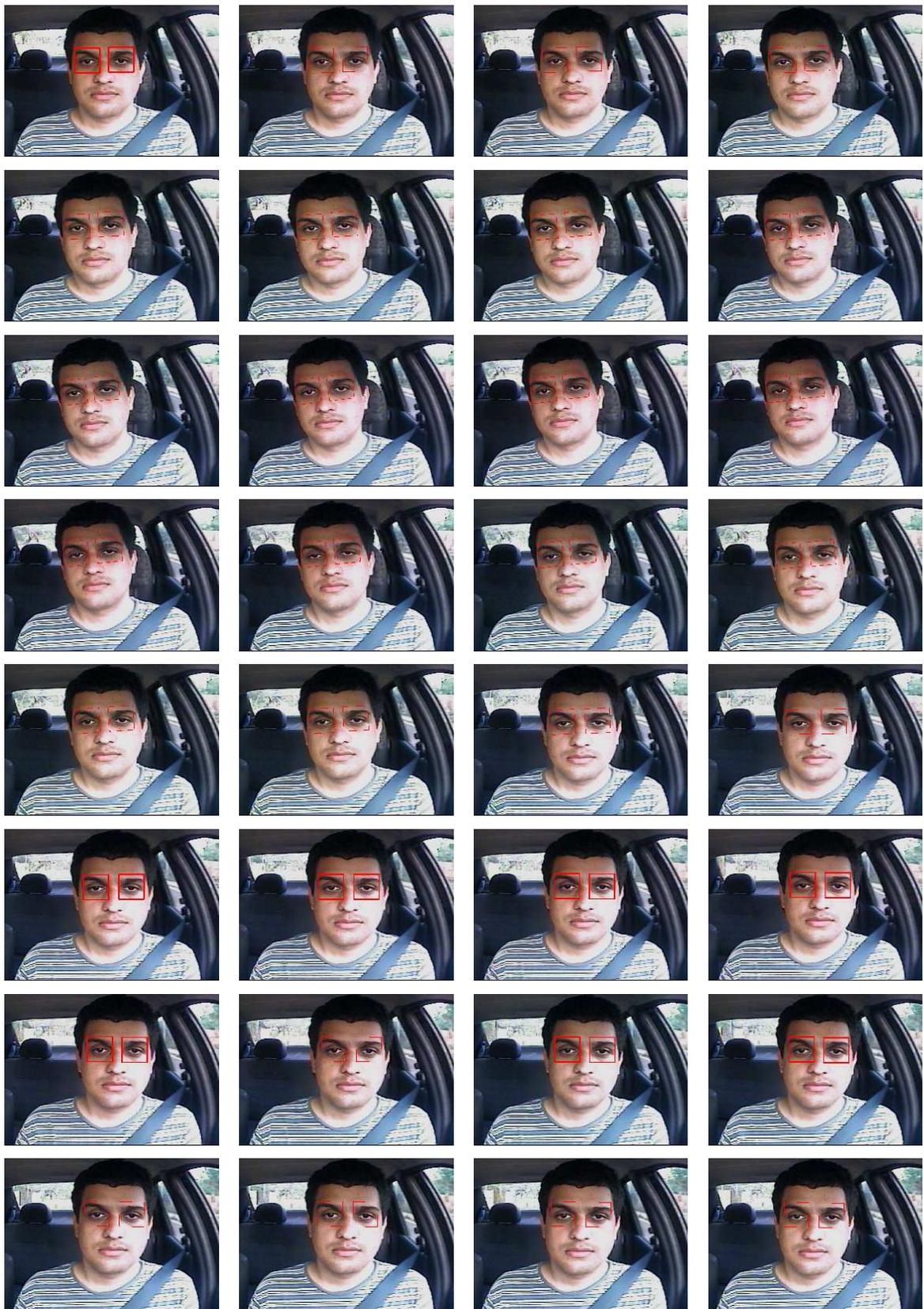


Figura 4.2: Resultado do algoritmo utilizado de detecção de olhos.

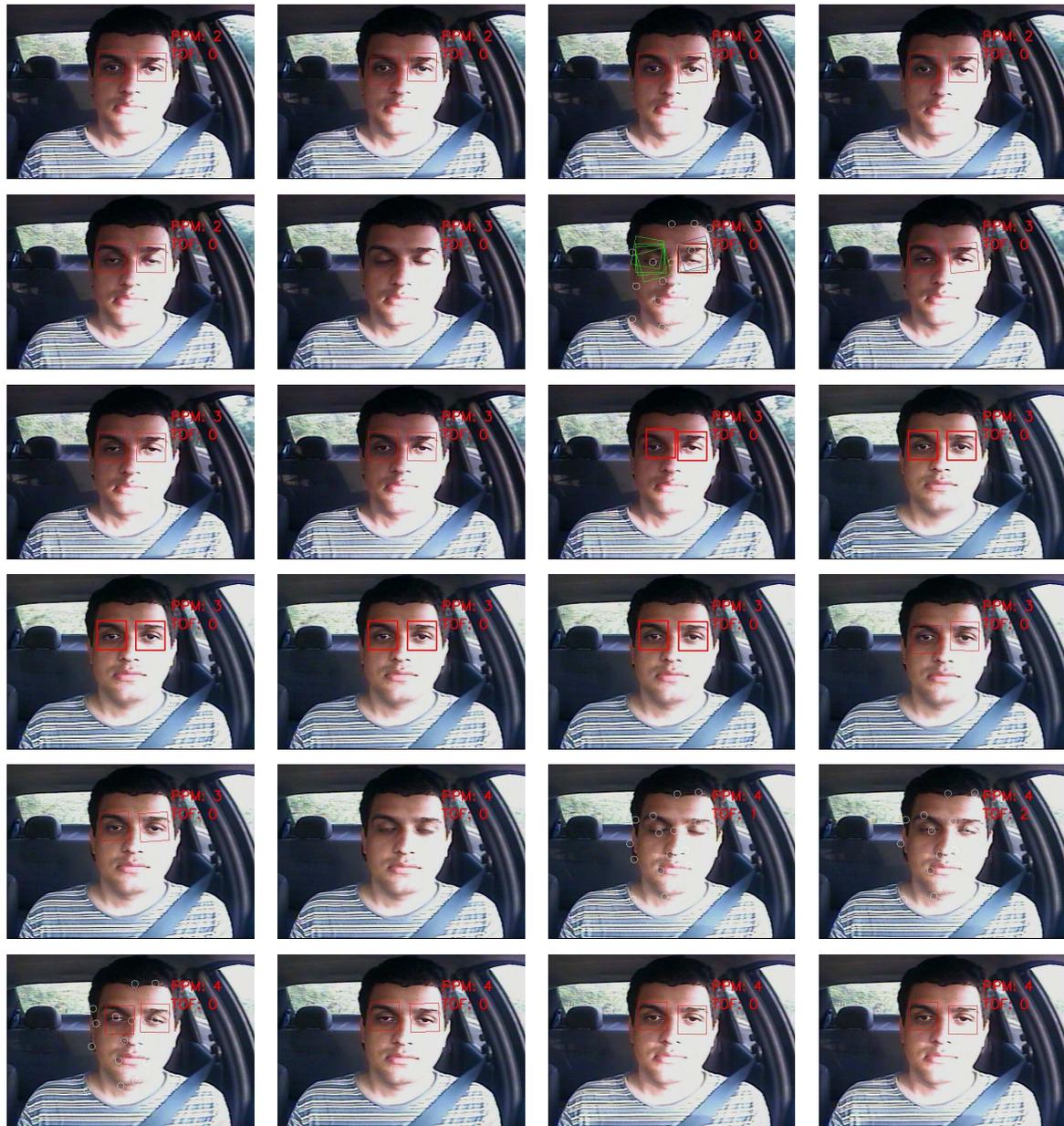


Figura 4.3: Resultado da detecção de piscada e cálculo das métricas de desatenção.

5 Conclusão

Este trabalho apresentou uma metodologia automatizada para detectar o cansaço e a fadiga de motoristas de veículos automotores baseado nas propriedades físicas dos olhos, e assim tentar minimizar a quantidade de acidentes de trânsito. Esta metodologia é capaz de fazer essa detecção de forma automática e de processar as informações da situação atual do motorista, fornecendo resposta em tempo real de acordo com o seu comportamento, sem contudo, atrapalhar o motorista na sua forma natural de dirigir.

As contribuições feitas com relação aos trabalhos relacionados analisados durante o desenvolvimento da metodologia envolvem a utilização de mecanismos de normalização de iluminação para o contorno de alguns problemas encontrados em situações reais devido a variação da iluminação e a utilização de mecanismos que permitem corrigir as distorções de orientação da região de interesse (olhos) para aumentar o desempenho de classificação da metodologia, retirando a necessidade de treinamento de diversas orientações de olhos.

Os resultados foram bastante promissores. Nos testes efetuados foi possível identificar e monitorar o estado dos olhos em média 89% dos quadros dos vídeos de teste. Esse é um indicador da robustez da metodologia proposta, pois essa taxa foi alcançada permitindo a movimentação natural do motorista e mudanças de iluminação total ou parcial.

Entretanto algumas dificuldades foram encontradas. Em alguns casos a iluminação intensa do Sol em combinação com propriedades do rosto do motorista impossibilitam o funcionamento do método, pois as regiões de captura apresentam apenas a cor branca ou uma grande região de alto brilho especular, contudo a falha nesses casos pode não ser muito significativa, pois a capacidade visual do motorista estará dificultada devido ao ofuscamento.

Devido as etapas de verificação de simetria, o sistema falha em situações onde a busca pela região dos olhos precisa ser feita quando a cabeça do motorista está virada para os lados. Nesses casos o sistema identificará os olhos do motorista como fechados. Não se trata de uma falha grave, pois os casos de ocorrência são quando os olhos do

motorista estão apontados para direções que não permitem a visualização da estrada. O sistema gerará uma mensagem de aviso. Mas os índices de verificação do estado de atenção do motorista são prejudicados.

Por último devido às buscas globais por simetria a capacidade de processamento do sistema no equipamento utilizado para fazer os testes é, na fase de busca global, 7 quadros por segundo, que não atende atualmente a capacidade de processamento em tempo real do sistema.

Para contornar algumas limitações e dificuldades encontradas, são propostos como trabalhos futuros:

- Melhora do desempenho do sistema utilizando unidades de processamento mais modernas com vários núcleos de processamento, o que pode melhorar significativamente o desempenho pois as etapas mais custosas do sistema, como busca por simetria de todos os candidatos a olhos e validação dos candidatos restantes são paralelizáveis.
- Utilização de câmeras infra-vermelho para utilização do sistema em baixa iluminação.
- Utilização de mais de uma câmera para diferenciar olhos fechados de olhos que não estão direcionados para frente .
- Testes com um maior amostragem.

REFERÊNCIAS

- Adini Y., Moses Y., e Ullman S. (1997). Face recognition: the problem of compensating for changes in illumination direction. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 19, pages 721–732. IEEE Computer Society.
- Almeida J. D. S. d. (2007). *Detecção de Falhas em Dados Sísmicos usando Máquinas de Suporte Vetorial*. Monografia , Curso de Ciência da Computação da Universidade Federal do Maranhão.
- Atherton T. J. e Kerbyson D. J. (1999). Size invariant circle detection. In *Image Vision Comput 17*, volume 17, pages 795–803. Elsevier Ltd.
- Bao D., Yang Z., e Song Y. (2007). Projection function for driver fatigue monitoring with monocular camera. In *Proceedings of the 2007 ACM symposium on Applied computing*, pages 82–83. ACM.
- Bisognin G. (2007). Utilização de máquinas de suporte vetorial para predição de estruturas terciárias de proteínas. In *Dissertação de Mestrado em Computação Aplicada*. Universidade do Vale do Rio dos Sinos.
- Chaves A. C. F. (2006). Extração de regras fuzzy para máquinas de vetor de suporte (svm) para classificação em múltiplas classes. In *Tese de Doutorado*. Pontifícia Universidade Católica do Rio de Janeiro.
- Cristianini N. e Saxe-Taylor J. (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press.
- Davies E. R. (2004). *Machine Vision: Theory, Algorithms, Practicalities*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- DETRAN-RS (2008). De olhos bem abertos. disponível em: <http://www.detran.rs.gov.br/clipping/20080308/01.htm>, Acessado em: 16 de junho de 2010.

- Dinges D. e Grace R. (1998). Perclos: A valid psychophysiological measure of alertness as assessed by psychomotor vigilance. In *US Department of Transportation, Federal Highway Administration, TechBrief, FHWA-MCRT-98-006*. Elsevier Ltd.
- D´Orazio T., Guaragnella C., Leo M., e Distante A. (2007). A visual approach for driver inattention detection. In *Pattern Recognition 40*, pages 2341–2355. Elsevier Ltd.
- Esquef I. A., Albuquerque M. P. D., e Albuquerque M. P. D. (2003). Processamento Digital de Imagens. *Artigo de Divulgação Científica*.
- Fan X., Yin B., e Sun Y. (2008). Nonintrusive driver fatigue detection. In *IEEE International Conference on Networking, Sensing and Control*, pages 905–910. IEEE Computer Society.
- Feng G. C. e Yuen P. C. (1998). Variance projection function and its application to eye detection for human face recognition. In *Pattern Recogn. Lett.*, volume 19, pages 899–906. Elsevier Science Inc.
- Foley J. D. (1996). *Computer Graphics - Principles and Practice*. Addison-Wesley.
- Gonzalez R. e Wintz P. (2002). *Digital Image Processing*. Prentice-Hall.
- Graps A. (1995). Wavelets for Computer Graphics: A Primer, Part 1. In *Computing in Science and Engineering*, pages 50–61.
- Haykin S. e Engel P. (2001). *Redes neurais: Principios e pratica*. Bookman.
- Hsu R.-L., Abdel-Mottaleb M., e Jain A. K. (2002). Face detection in color images. In *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 696–706. IEEE Computer Society.
- Intel (2000). Open source computer vision library, disponível em <http://sourceforge.net/projects/opencvlibrary/>, acessado em: 02 de dezembro de 2009.
- Jiao F. e He G. (2007). Real-time eye detection and tracking under various light conditions. In *Data Science Journal, Volume 6, Supplement, 29 September 2007*, pages 636–640.
- Li S. (2002). Multi-faces location in color image. *Concordia University, thesis*.

- Liatsis P. e Nazarboland M. A. (2002). Microarray image analysis. In *Proceedings of the 9th International Workshop on Systems, Signals and Image Processing*, pages 515–521. World Scientific Publishing.
- Luhandjula T. K. (2007). Closed eye detection for fatigue monitoring. In *fulfilment of the requirements for the degree of Magister Technologiae*. French South African Technical Institute in Electronics.
- Maulin Gandhi M. D. L. e Bhattacharyya J. (2004). Image normalization for illumination compensation in facial images. *Department of Electrical and Computer Engineering, Center for Intelligent Machines*.
- Savas Z. (2005). Real-time detection and tracking of human eyes in video sequences. In *Master Thesis*. School Of Natural And Applied Sciences Of Middle East Technical University.
- Silva R. D., Minetto R., e Pedrini H. . (2006). Segmentação de imagens médicas utilizando transformada de wavelets. In *Revista Eletrônica de Iniciação Científica*, volume 6, pages 18–27.
- Sonka M., Hlavac V., e Boyle R. (2008). *Image Processing Analysis and Machine Vision*. Chapman-Hall.
- Stollnitz E. J., DeRose, Tony D. S., e H. D. (1995). Wavelets for Computer Graphics: A Primer, Part 1. In *IEEE Computer Graphics and Applications*, volume 15, pages 76–84. IEEE Computer Society.
- Su M. C., Hsiung C. Y., e Huang D. Y. (2006). A simple approach to implementing a system for monitoring driver inattention. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 429–433. IEEE Computer Society.
- Tomasi C. (2007). *Convolution, Smoothing, and Image Derivatives*. Class handout.
- Vuçini E., Gökmen M., e Gröller M. E. (2007). Face recognition under varying illumination. In *15th WSCG 2007*, pages 57–64. University of West Bohemia.
- Yuen H. K., Princen J., Illingworth J., e Kittler J. (1990). Comparative study of hough transform methods for circle finding. volume 8, pages 71–77, Newton, MA, USA. Butterworth-Heinemann.

Zhou Z.-H. e Geng X. (2004). Projection functions for eye detection. In *Pattern Recognition*, volume 37, pages 1049–1056.