

UNIVERSIDADE FEDERAL DO MARANHÃO  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA  
CURSO DE CIÊNCIA DA COMPUTAÇÃO

SAMUEL VINGREMBERG CHAVES SOARES DA SILVA

RASTREAMENTO OCULAR UTILIZANDO FILTRO DE CORRELAÇÃO

SÃO LUÍS

2019

SAMUEL VINGREMBERG CHAVES SOARES DA SILVA

RASTREAMENTO OCULAR UTILIZANDO FILTRO DE CORRELAÇÃO

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão como parte dos requisitos para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. João Dallyson Sousa de Almeida

SÃO LUÍS

2019

---

Samuel Vingremberg Chaves Soares da Silva

Rastreamento ocular utilizando filtro de correlação/ Samuel Vingremberg  
Chaves Soares da Silva. - 2019.

45 p.

Orientador: Prof. Dr. João Dallyson Sousa de Almeida

Monografia (Graduação) - Curso de Ciência da computação, Universidade Federal  
do Maranhão, São Luís - MA, 2019.

1. rastreamento, filtro de correlação, olhos.

I. Sousa de Almeida, João Dallyson. II. Doutor.

---

## **TERMO DE APROVAÇÃO**

**SAMUEL VINGREMBERG CHAVES SOARES DA SILVA**

### **RASTREAMENTO OCULAR UTILIZANDO FILTRO DE CORRELAÇÃO**

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão como parte dos requisitos para a obtenção do grau de Bacharel em Ciência da Computação, pela seguinte banca examinadora:

Aprovada em 29 de Julho de 2019

---

**Prof. Dr. João Dallyson Sousa de Almeida**  
Orientador  
Universidade Federal do Maranhão

---

**Prof. Dr. Anselmo Cardoso de Paiva**  
Examinador  
Universidade Federal do Maranhão

---

**Profa. Dra. Simara V. da Rocha**  
Examinadora  
Universidade Federal do Maranhão

São Luís, MA  
2019

## **AGRADECIMENTOS**

Agradeço a Deus, à minha família e aos meus amigos. Razão de tudo o que sou.

À minha namorada, Andressa de Carvalho, que sempre acredita em mim e tem sido o meu apoio.

Ao meu orientador, Prof. Dr. João Dallyson Sousa de Almeida por sempre estar disposto a me ajudar, mesmo quando eu não mereço.

À coordenadora do curso de Ciência da Computação na UFMA, Profa. Dra. Simara V. da Rocha, por fazer mais do que sua obrigação e se preocupar com o futuro dos alunos do curso.

Aos meu colegas de curso e demais professores, que me acompanharam nessa jornada.

*"It's just a matter of time."*

## RESUMO

O rastreamento do olho é uma tarefa computacional que pode ser usada para os mais diversos fins, desde analisar o comportamento do usuário até controle realizado pelo movimento dos olhos. Nos últimos anos, rastreadores baseados em filtro de correlação estão sendo amplamente desenvolvidos e testados, no intuito de rastreamentos mais precisos e menos sensíveis à mudanças de iluminação e escala. Neste trabalho, propomos um método de rastreamento do olho utilizando filtro de correlação KCF para um rastreamento rápido e eficiente. O método proposto obteve resultados satisfatórios, que sinalizam o potencial do uso do filtro de correlação para o rastreamento dos olhos.

**Palavras-chaves:** rastreamento, filtro de correlação, olhos.

## ABSTRACT

Eye tracking is a computational task that can be used for a wide range of purposes, from analyzing user behavior to controlling eye movement. In recent years, correlation filter-based trackers have been extensively developed and tested for more accurate tracking and less sensitivity to changes in light and scale. In this work, we propose an eye tracking method using KCF correlation filter for fast and efficient tracking. The proposed method obtained satisfactory results, which indicate the potential of using the correlation filter for eye tracking.

**Key-words:** tracking, correlation filter, eyes.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de utilização da equalização de histograma . . . . .	17
Figura 2 – Representação da imagem integral . . . . .	18
Figura 3 – Reta em função do raio e do ângulo . . . . .	19
Figura 4 – Transformada de Hough para círculos . . . . .	20
Figura 5 – Exemplos de características Haar . . . . .	21
Figura 6 – Cálculo da área do retângulo através da imagem integral . . . . .	22
Figura 7 – Cascata de rejeição utilizada no classificador Viola-Jones . . . . .	23
Figura 8 – Deslocamento cíclicos 1d. . . . .	28
Figura 9 – Deslocamento cíclicos 2d. . . . .	28
Figura 10 – Filtro de correlação e sua saída . . . . .	30
Figura 11 – Esquema do método proposto . . . . .	31
Figura 12 – Esquema da fase de detecção da face e dos olhos . . . . .	33
Figura 13 – Detecção dos olhos . . . . .	39
Figura 14 – Resultado da detecção dos olhos . . . . .	39
Figura 15 – Rastreamento do olho . . . . .	40
Figura 16 – Teste do rastreamento do olho . . . . .	40
Figura 17 – Teste do rastreamento do olho com variação de escala . . . . .	41
Figura 18 – Teste do rastreamento do olho com variação de rotação . . . . .	41
Figura 19 – Resultado do rastreamento . . . . .	42
Figura 20 – Resultado dos sub-conjuntos que possuem variação de rotação e escala . . . . .	42
Figura 21 – Resultado dos sub-conjuntos que possuem variação iluminação . . . . .	43

## LISTA DE ABREVIATURAS E DE SIGLAS

**ASEF** Average of Synthetic Exact Filters

**DCF** Dual Correlation Filter

**DFT** Discrete Fourier Transform

**FN** Falso Negativo

**FP** Falso Positivo

**FT** Fourier Transform

**KCF** Kernelized Correlation Filter

**MOSSE** Minimum Output Sum of Squared Error

**RAM** Random Access Memory

**TLD** Tracking-learning-detection

**UMACE** Unconstrained Minimum Average Correlation Energy

**VN** Verdadeiro Negativo

**VP** Verdadeiro Positivo

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>12</b>
1.1	OBJETIVOS	13
1.1.1	Objetivo geral	13
1.1.2	Objetivos específicos	13
1.2	TRABALHOS RELACIONADOS	13
1.3	ORGANIZAÇÃO DO TRABALHO	14
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>16</b>
2.1	PROCESSAMENTO DE IMAGENS	16
2.1.1	Representação digital da imagem	16
2.1.2	Equalização de histograma	16
2.1.3	Imagem Integral	18
2.2	TRANSFORMADA DE HOUGH	18
2.3	CLASSIFICADOR EM CASCATA HAAR CASCADES	21
2.4	FUNÇÃO GAUSSIANA	22
2.5	TRANSFORMADA DE FOURIER	23
2.5.1	Transformada discreta de Fourier (1-D)	23
2.5.2	Transformada discreta de Fourier (2-D)	24
2.6	MODELO DA REGRESSÃO LINEAR	25
2.7	DESLOCAMENTOS CÍCLICOS	27
2.8	MATRIZES CIRCULANTES	28
2.9	FILTRO DE CORRELAÇÃO	29
<b>3</b>	<b>MÉTODO PROPOSTO</b>	<b>31</b>
3.1	PRÉ-PROCESSAMENTO	32
3.2	DETECÇÃO DA FACE E DO OLHO	33
3.3	RASTREAMENTO DO OLHO	34
3.3.1	Algoritmo e parametrização do filtro	35
<b>4</b>	<b>TESTES E RESULTADOS</b>	<b>37</b>
4.1	MÉTRICAS	37
4.2	RESULTADOS DA DETECÇÃO DO OLHO	38
4.3	RESULTADOS DO RASTREAMENTO DO OLHO	39
4.3.1	Testes e resultados	41
<b>5</b>	<b>CONCLUSÃO</b>	<b>44</b>

**REFERÊNCIAS . . . . . 45**

## 1 INTRODUÇÃO

O rastreamento visual tem muitas aplicações práticas na visão computacional, como segurança em vídeo, interface humano-computador e percepção robótica. Quando um alvo está localizado em um quadro de um vídeo, geralmente é útil rastrear esse objeto em quadros subsequentes. Cada quadro no qual o alvo é rastreado com sucesso fornece mais informações sobre a identidade e a atividade do alvo. Esse alvo pode ser um carro, uma pessoa, uma face ou até mesmo os olhos.

As tecnologias e processos envolvidos com o rastreamento dos olhos originalmente tiveram motivações clínicas, sendo aplicados no estudo de disfunções e lesões nos olhos, distúrbios de leitura e outros problemas oftalmológicos. Algum tempo depois, esta técnica começou a ser empregada em outras disciplinas, como na psicologia e neurologia, no desenvolvimento e validação de modelos cognitivos, no estudo da percepção, e na análise de tarefas comuns, como dirigir um veículo. (GIANNOTTO, 2009)

Segundo Tobii (2019), o rastreamento do olhar (ou rastreamento ocular), é um termo geral para técnicas de medição do ponto observado – onde você está olhando – ou para determinar a posição do olho em relação à cabeça. Ou seja, existem basicamente dois fins para os quais se realizam o rastreamento ocular: para se determinar o ponto que está sendo visto pelo usuário ou para se determinar a posição dos olhos em relação à cabeça. Um sistema capaz de realizar o rastreamento do olhar é conhecido pelo nome de *eye tracker*. As aplicações do rastreamento ocular podem ser classificadas em diagnósticas ou interativas. As diagnósticas oferecem evidências objetivas e quantitativas na forma de dados fornecidos pelo rastreador, enquanto as interativas fornecem uma resposta ao usuário a partir dos dados obtidos.

O rastreamento dos olhos recebeu muita atenção nos últimos anos. Várias estratégias robustas de rastreamento foram propostas para tolerar mudanças na aparência do alvo e rastrear alvos através de movimentos complexos. Exemplos recentes incluem: *Incremental Visual Tracking (IVT)* (ROSS et al., 2008), *Robust Fragments-based Tracking (FragTrack)* (ADAM; RIVLIN; SHIMSHONI, 2006), e *Graph Based Discriminative Learning (GBDL)* (ZHANG et al., 2007). Embora eficazes, essas técnicas não são simples. Elas geralmente incluem modelos complexos de aparência e / ou algoritmos de otimização e, como resultado, se esforçam para alcançar os 25 a 30 quadros por segundo produzidos por muitas câmeras modernas.

A proposta desse trabalho consiste no desenvolvimento de uma metodologia de rastreamento dos olhos baseada no filtro de correlação usando o método de

rastreamento de objetos proposto por Henriques et al. (2014).

## 1.1 OBJETIVOS

Nesta seção estão contidos os objetivos geral e específicos deste trabalho.

### 1.1.1 Objetivo geral

De acordo com o contexto apresentado, o objetivo geral deste trabalho consiste em desenvolver um método de rastreamento dos olhos baseado no filtro de correlação, capaz de seguir o movimento dos olhos sem uma alta demanda computacional, operando, assim, a um alto índice de quadros por segundo.

### 1.1.2 Objetivos específicos

Para a concretização do objetivo geral, alguns objetivos específicos foram realizados:

- Aquisição de vídeo da face;
- Detecção da face e dos olhos;
- Extração a região de interesse para o rastreamento;
- Rastreamento da região de interesse utilizando o rastreador proposto neste trabalho;
- Avaliação dos resultados do rastreador.

## 1.2 TRABALHOS RELACIONADOS

Embora ainda não sejam muito conhecidos, alguns trabalhos que apresentam métodos de rastreamento com emprego do filtro de correlação têm demonstrado o quanto essa metodologia pode ser promissora para este fim.

O componente principal da maioria dos rastreadores modernos é um classificador discriminativo, com a tarefa de distinguir entre o alvo e o ambiente ao redor. Para lidar com mudanças naturais de uma imagem (como iluminação e escala), esse classificador é normalmente treinado com amostras do alvo redimensionados e transladados. Esses conjuntos de amostras geralmente estão repletos de redundâncias, o que acaba gerando um grande consumo computacional.

Os filtros de correlação já são utilizados há algum tempo para a tarefa de rastreamento, mas os mais antigos e simples usam modelos frágeis e geralmente

falham quando aplicados ao rastreamento (DANELLJAN et al., 2014). Abordagens mais modernas, como ASEF (BOLME; DRAPER; BEVERIDGE, 2009) e UFACE (MAHALANOBIS et al., 1994), têm melhor desempenho, mas suas necessidades de treinamento são pouco adequadas ao rastreamento. O rastreamento visual requer filtros robustos para serem treinados a partir de um único quadro e dinamicamente adaptados à medida que a aparência do objeto alvo é alterada.

Com base nessas observações, Henriques et al. (2014) propuseram um modelo analítico para conjuntos de dados de milhares de amostras transladadas. Ao mostrar que a matriz de dados resultante é circulante, Henriques et al. (2014) conseguiu diagonalizá-la com a transformada discreta de Fourier ou *discrete Fourier transform*, reduzindo o armazenamento e a computação em várias ordens de magnitude. Por regressão linear, esta formulação converge para um filtro de correlação. Para a regressão do *kernel*, no entanto, foi derivado um novo filtro: o filtro de correlação de *kernel* ou *Kernelized Correlation Filter* (KCF), que, ao contrário de outros algoritmos de *kernel*, tem exatamente a mesma complexidade de seu equivalente linear.

Com base nisso, Henriques et al. (2014) também propuseram uma rápida extensão multicanal de filtros de correlação linear, através de um *kernel* linear, que chamou de filtro de correlação dual ou *Dual Correlation Filter* (DCF). Tanto o KCF quanto o DCF superam os principais rastreadores, como *Struck* (HARE et al., 2015) ou TLD (KALAL; MIKOLAJCZYK; MATAS, 2011), em um *benchmark*<sup>1</sup> de 50 vídeos, apesar de rodarem a centenas de quadros por segundo.

Além do trabalho de Henriques et al. (2014), também podemos citar a metodologia proposta por Bolme, Beveridge et al. (2010), onde é apresentado um novo tipo de filtro de correlação, um filtro de Soma Mínima de Saída de Erro Quadrático ou *Minimum Output Sum of Squared Error* (MOSSE), que produz filtros de correlação estáveis quando inicializado usando um único quadro. Um rastreador baseado em filtros MOSSE é robusto para variações de iluminação, escala, pose e deformações não rígidas enquanto opera a 669 quadros por segundo.

O problema de rastreamento dos olhos não foge do escopo do rastreamento de objetos, sendo uma especialização deste. Ainda que se torne um problema mais específico, este trabalho demonstra, a possibilidade da aplicação do método KCF no rastreamento do olho.

### 1.3 ORGANIZAÇÃO DO TRABALHO

O trabalho está organizado da seguinte forma:

No Capítulo 2, Fundamentação Teórica, segue informações importantes para o

---

<sup>1</sup> Um padrão ou ponto de referência contra o qual as coisas podem ser comparadas ou avaliadas.

contexto e entendimento do trabalho, tais como explicações sobre Filtro de correlação, Haar Cascade e alguns conceitos abordados em Processamento de Imagens.

No Capítulo 3, Método Proposto, explica-se a metodologia utilizada como ponto de partida para o desenvolvimento desse trabalho.

No Capítulo 4, Testes e Resultados, são apresentados os testes e os resultados alcançados na aplicação da metodologia, junto como uma análise de seu desempenho.

No Capítulo 5, Conclusão, apresenta-se a conclusão do trabalho. Nela está contida uma retrospectiva do que foi apresentado, uma discussão dos resultados obtidos e também sugestões para o prosseguimento e aprimoramento desse trabalho.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo aborda os conceitos matemáticos e de processamento de imagens necessários para um bom entendimento e desenvolvimento do trabalho.

### 2.1 PROCESSAMENTO DE IMAGENS

Nesta subseção, seguem alguns conceitos básicos, porém importantes, sobre processamento de imagens, que serão úteis no decorrer do trabalho.

#### 2.1.1 Representação digital da imagem

A imagem pode ser representada como uma função bidimensional  $f$  e a amplitude  $f(x, y)$  é o valor de sua intensidade no ponto  $(x, y)$ . Denomina-se de imagem digital quando  $x$  e  $y$ , assim como a intensidade de  $f$  possuem valores finitos e discretos. Computacionalmente, uma imagem também pode ser representada por uma matriz de tamanho  $M \times N$ , e cada célula dessa matriz é chamada de *pixel*, abreviação do inglês para *picture element*. A quantidade de *bits* utilizada para representar este valor é chamada profundidade do *pixel*, quanto maior a profundidade, maior o número de diferentes intensidades que é possível representar. Em imagens coloridas os *bits* em cada *pixel* são interpretados de acordo com o sistema de cor que se encontra codificada a imagem.

As imagens a serem utilizadas neste trabalho são em escala de cinza, e as mesmas foram capturadas através da webcam utilizando a biblioteca OpenCV.

#### 2.1.2 Equalização de histograma

O conceito de histograma, segundo Marques Filho e Neto (1999), é um conjunto de números indicando o percentual de *pixels* naquela imagem que apresentam um determinado nível de cinza. Estes valores são normalmente representados por um gráfico de barras que fornece para cada nível de cinza o número (ou o percentual) de *pixels* correspondentes na imagem. Através da visualização do histograma de uma imagem obtemos uma indicação de sua qualidade quanto ao nível de contraste e quanto ao seu brilho médio (se a imagem é predominantemente clara ou escura).

$$P_r(r_k) = \frac{n_k}{n} \quad (2.1)$$

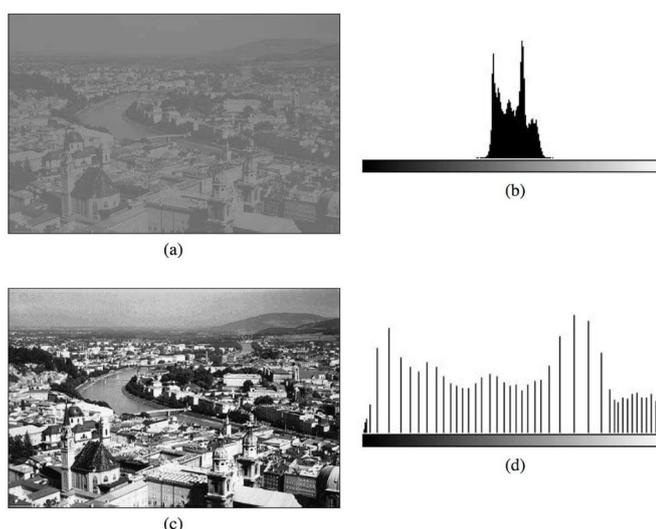
onde  $0 \leq P_r \leq 1$ ,  $r_k$  é o  $k$ -ésimo nível de intensidade em uma faixa  $[0 - L)$ ,  $n_k$  é o número de ocorrências de *pixels* cujo valor de intensidade é  $r_k$ , e  $n$  é o número total de

*pixels* na imagem. Em imagens em nível de cinza em que a profundidade do *pixel* é 8 *bits*,  $L$  assume o valor de 256.

A equalização do histograma é uma das técnicas mais utilizadas na literatura quando deseja-se obter um realce genérico de contrastes. Tendo como finalidade redistribuir os valores dos tons de cinza, de modo a obter um histograma mais uniforme.

Um exemplo de histograma é apresentado na Figura 1, que apresenta em (a) a imagem original e seu respectivo histograma em (b). Como dito anteriormente é possível, através do histograma, observar que a imagem possui um baixo contraste, como também visualizar qual é seu brilho médio. Em (c) tem-se o resultado da imagem equalizada e seu respectivo histograma (d), agora com uma distribuição mais uniforme devido ao realce no contraste.

FIGURA 1 – Exemplo de utilização da equalização de histograma



Fonte: Marques Filho e Neto (1999)

Apesar de sua utilização ampla para aprimoramento de contraste, a equalização de histograma tem como a principal característica o fato de não ser parametrizada. Podendo a nova imagem equalizada ser obtida pela Equação 2.2, onde  $T(r_k)$  é a nova intensidade para o *pixel* de grau de intensidade  $k$ .

$$T(r_k) = (L - 1) \cdot \sum_{i=0}^k p_r(r_i) \quad (2.2)$$

Neste trabalho, a equalização de histograma é utilizada para melhorar possíveis imagens com baixo contraste. A equalização do histograma auxilia na fase de detecção da face e dos olhos.

### 2.1.3 Imagem Integral

A Imagem Integral (*Integral Image*) é uma técnica de programação dinâmica por realizar uma série de operações armazenando os resultados para reuso futuro. A Imagem Integral consiste em uma matriz de acumuladores que armazena a soma da intensidade dos *pixels*, permitindo rápidas operações em sub-regiões de uma imagem. O que é de grande utilidade em processamento de imagens uma vez que cálculos como média e soma podem ser realizados eficientemente, reduzindo o tempo de processamento de imagens.

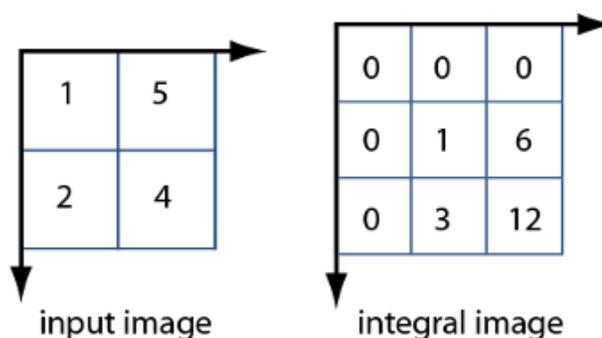
A Imagem Integral também é comumente utilizada para o cálculos das extrações de características baseadas em *Wavelets* de Haar, como proposto por Viola e Jones (2001).

A Imagem Integral no *pixel* de posição  $(x, y)$  contém a soma dos *pixels* acima e a esquerda de  $(x, y)$ , como também o próprio valor de  $(x, y)$  na imagem original. Matematicamente é expresso pela Equação 2.3.

$$I(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (2.3)$$

Por motivos computacionais, a imagem integral possui dimensões  $N + 1 \times M + 1$ , com a primeira linha e coluna com valores preenchidas com zeros. A Figura 2 demonstra um exemplo desta operação.

FIGURA 2 – Representação da imagem integral



Fonte: Marques Filho e Neto (1999) <sup>1</sup>

## 2.2 TRANSFORMADA DE HOUGH

A transformada de Hough (BRADSKI; KAEHLER, 2013) é uma técnica utilizada em processamento de imagens para identificar formas geométricas básicas a partir de suas equações. Foi inicialmente proposta para a detecção de retas, mas foi estendida,

<sup>1</sup> Disponível em: <http://www.mathworks.com/help/vision/ref/integralimage.html>. Acesso em 23 de junho de 2019

posteriormente, para identificação de outras formas, tais como círculos e elipses. Na maioria dos casos a utilização de um detector de bordar pode ser utilizado em uma fase inicial de pré-processamento. Entretanto, devido a imperfeições na imagem ou no resultado obtido pela detecção, podem exibir uma borda com falhas, ou apresentar ruídos que se assimilam à formas não esperadas. A transformada de Hough trata este problema agrupando pontos que seriam de borda em grupos de objetos candidatos através de um processo de votação no plano dos parâmetros.

O caso mais simples da transformada de Hough é a detecção de retas. Uma reta pode ser descrita por  $y = ax + b$ , onde  $a$  é inclinação da reta em relação ao eixo  $x$  e  $b$  é o ponto onde a reta intercepta o eixo  $y$ . Uma reta pode ser representada pelo ponto  $(a, b)$  no plano dos parâmetros, porém representar retas verticais seria um problema, as retas  $x = 0$  e  $x = 1$ , por exemplo, não seriam distinguíveis. Por isso utiliza-se coordenadas polares para esse fim, conforme a equação

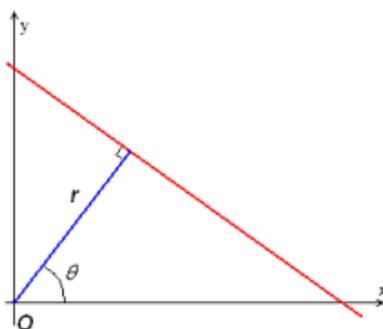
$$y = \left( -\frac{\cos \theta}{\sin \theta} \right) x + \left( \frac{r}{\sin \theta} \right) \quad (2.4)$$

Que pode ser reescrita como:

$$r = x \cos \theta + y \sin \theta \quad (2.5)$$

onde  $r$  é a menor distancia da reta para a origem, Figura 3.

FIGURA 3 – Reta em função do raio e do ângulo



Fonte: Wikipedia <sup>1</sup>

Associando, portanto, cada reta a um par  $(r, \theta)$  que é único, se  $\theta \in [0, \pi]$  e  $r \in \mathbb{R}$ .

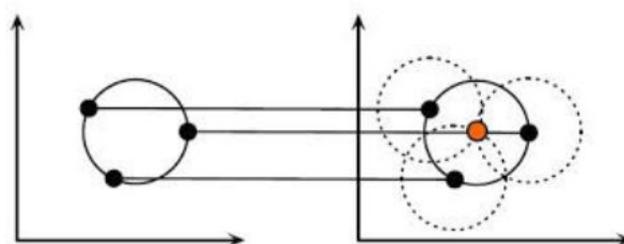
Cada um desses pontos  $(x_0, y_0)$  possui uma curva senoidal no plano  $(rx\theta)$  representando retas no formato  $(r, \theta)$ , que passam por cada um deles.

<sup>1</sup> Disponível em: [http://en.wikipedia.org/wiki/File:R\\_theta\\_line.GIF](http://en.wikipedia.org/wiki/File:R_theta_line.GIF). Acesso em 16 de junho de 2019

Se as curvas correspondentes a dois ou mais pontos se sobreporem, a localização onde elas se cruzam corresponde a reta que passam por aqueles pontos. Um acumulador é utilizado para identificar quantos pontos pertencem àquela reta. Certamente, o conjunto de pontos que fazem parte de um segmento de reta na imagem irá produzir senoidais que se cruzam nos parâmetros daquela reta.

A transformada de Hough para detecção de círculo funciona de maneira semelhante, conforme pode ser observado a partir de Figura 4.

FIGURA 4 – Transformada de Hough para círculos



Fonte: Página da ELETRONICA.ORG <sup>1</sup>

Para tal fim, seria usado, semelhantemente, as equações paramétricas da circunferência, dadas por:

$$x = a + r \cos \theta \quad (2.6)$$

$$y = b + r \sin \theta \quad (2.7)$$

onde  $a$  e  $b$  são as coordenadas do centro da circunferência respectivamente nos eixos  $x$  e  $y$ .

Porém, se for tratado dessa forma o acumulador, que antes possuía duas dimensões ou variáveis e podia ser entendido como um plano, teria agora que ser substituído por um acumulador de três dimensões: uma para  $x$ , uma para  $y$  e outra para o raio do círculo. O que implicaria em consumo excessivo de memória e de tempo de processamento. A implementação disponível no OpenCV evita esse problema por meio de um método chamado de gradiente de Hough, descrito por Bradski e Kaehler (2013).

A detecção de círculos implementada pelo OpenCV utiliza de uma função com vários parâmetros dentre eles: o limiar superior para a detecção de bordas do filtro de Canny, o limiar inferior utilizado é a metade deste; uma distancia mínima entre o centro dos círculos detectados, o tamanho mínimo do raio dos círculos detectados, assim como o tamanho máximo do raio.

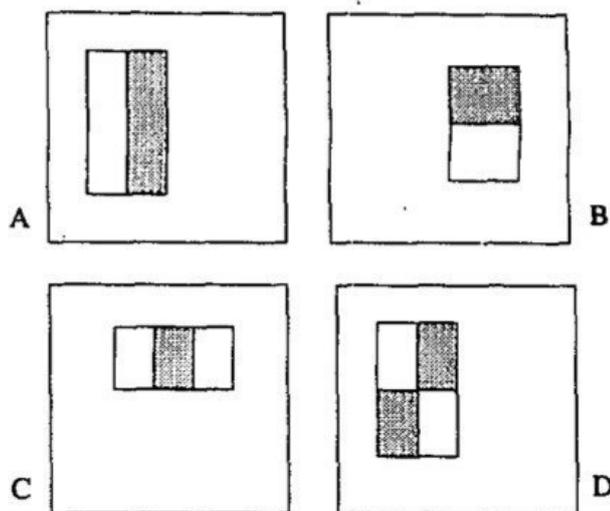
<sup>1</sup> Disponível em: <http://www2.eletronica.org/> Acesso em 16 de junho de 2019

Neste trabalho, a transformada de Hough foi utilizada depois da fase de detecção dos olhos para se estimar o centro da iris.

### 2.3 CLASSIFICADOR EM CASCATA HAAR CASCADES

O método de detecção descrito nessa sessão foi inicialmente proposto por Viola e Jones (2001). As características Haar ou *Haar features* utilizam um conjunto de características baseado nas *Haar Wavelets* que aplicam o uso de áreas retangulares. Viola e Jones (2001) adaptaram a ideia utilizando mais de uma área retangular, desenvolvendo as chamadas *Haar-like features*. Estas consistem em somar a intensidade de *pixels* uma dessas regiões, e utilizar como característica a diferença desses resultados para áreas adjacentes. A Figura 5 mostra as *Haar-like features* utilizadas por Viola e Jones (2001), onde A e B mostram características retangulares compostas por 2 retângulos, C possui três retângulos e D quatro retângulos. Em que, em cada uma dessas características, a região a que cada retângulo pertence é indicada pela sua cor de preenchimento nas figuras apresentadas.

FIGURA 5 – Exemplos de características Haar



Fonte: Viola e Jones (2001)

Para realizar a detecção em tempo real, a extração de milhares de características em sub-regiões de cada quadro em um vídeo seria algo muito custoso. Para a rápida computação dessas características, Viola e Jones (2001) propuseram também a utilização de uma Imagem Integral, que pode servir como uma tabela de consultas (*lookup table*).

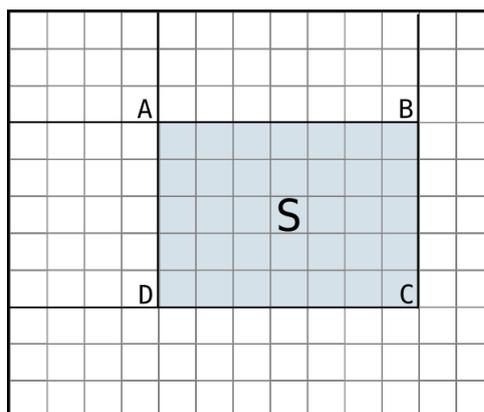
Assim, a soma dos valores da área sombreada na Figura 6, pode ser calculado

por:

$$Sum = I(C) + I(A) - I(B) - I(D) \quad (2.8)$$

sendo  $A, B, C, D$ , pontos pertencentes a imagem integral da Figura 5.

FIGURA 6 – Cálculo da área do retângulo através da imagem integral



Fonte: Viola e Jones (2001)

Dessa forma, a imagem integral é computada uma única vez, e para cada característica Haar-like são necessárias 6 consultas para características com 2 retângulos, 8 para 3 retângulos e 9 consultas para características com 4 retângulos.

Mesmo com a eficiência obtida pelo uso da Imagem Integral, a quantidade dessas características a serem extraídas são muitas. Em uma sub-região de 24 x 24 *pixels* existem em torno de 160000 delas (VIOLA; JONES, 2001). Por ser alto o número de características a serem avaliadas, para uma detecção em tempo real utiliza-se uma variação do algoritmo de aprendizagem AdaBoost, que seleciona as melhores características a serem utilizadas para treinar os classificadores.

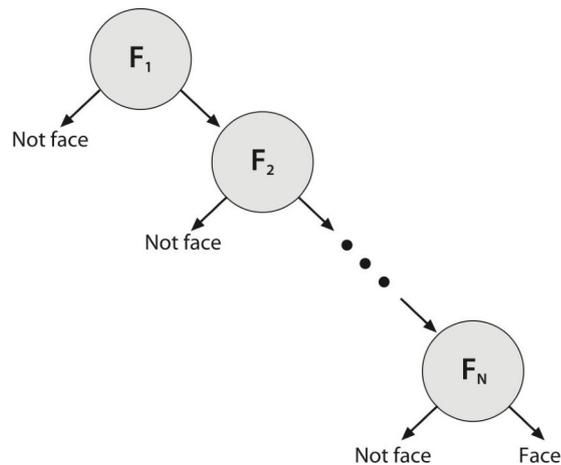
Para a detecção, os classificadores fortes são organizados em formato de cascata de acordo com sua complexidade. Se em algum estágio um classificador rejeita uma sub-janela em uma análise, aquela região não é passada ao próximo classificador, e a busca pelo objeto passa a ser feita em uma nova sub-janela. O classificador tem, portanto, o formato de uma árvore degenerada, conforme a Figura 7.

Neste trabalho, foi utilizado o classificador em cascata para a detecção da face e dos olhos.

## 2.4 FUNÇÃO GAUSSIANA

Em probabilidade e estatística, função gaussiana é uma das distribuições de probabilidade mais utilizadas para modelar fenômenos naturais. Isso se deve ao fato de que um grande número de fenômenos naturais apresenta sua distribuição

FIGURA 7 – Cascata de rejeição utilizada no classificador Viola-Jones



Fonte: Viola e Jones (2001)

de probabilidade muito próxima da gaussiana. A função gaussiana é ligada a vários conceitos matemáticos como movimento browniano, ruído branco, entre outros. A função gaussiana também é chamada distribuição normal, distribuição de Gauss ou distribuição de Laplace–Gauss, em referência aos matemáticos, físicos e astrônomos francês Pierre–Simon Laplace (1749 – 1827) e alemão Carl Friedrich Gauss (1777 – 1855). (ASTRONOMY, 2008)

Uma função gaussiana tem a forma:

$$y(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\left(\frac{x-\mu}{2\sigma^2}\right)} \quad (2.9)$$

onde  $\sigma$  é o desvio padrão, e  $\mu$  é a expectância de  $x$ .

Neste trabalho, a função gaussiana é utilizada para redistribuir os valores dos *pixels* da sub-imagem do olho antes da fase de treinamento do filtro de correlação,

## 2.5 TRANSFORMADA DE FOURIER

Esta seção apresenta de forma sucinta os principais conceitos e propriedades da transformada de Fourier (FT), (MARQUES FILHO; NETO, 1999) ferramenta matemática obrigatória para a especificação e projeto de filtros no domínio da frequência.

### 2.5.1 Transformada discreta de Fourier (1-D)

Supondo que uma função contínua  $f(x)$  seja discretizada, produzindo a sequência  $f(x_0), f(x_0 + \Delta x), f(x_0 + 2\Delta x), \dots, f(x_0 + [N - 1]\Delta x)$  através da amostragem de  $N$  pontos, espaçados entre si de  $\Delta x$ .

Denominando a função discreta  $f(n)$ ,

$$f(n) = f(x_0 + n\Delta x) \quad (2.10)$$

onde  $n$  pode assumir os valores discretos  $0, 1, 2, \dots, N-1$ , podemos interpretar a seqüência  $f(0), f(1), f(2), \dots, f(N-1)$  como qualquer seqüência de  $N$  amostras consecutivas do sinal original contínuo  $f(x)$ , uniformemente espaçadas.

A transformada de Fourier de  $f(n)$  será:

$$F(u) = \frac{1}{N} \sum_{n=0}^{N-1} f(n) \exp \left[ \frac{-2j\pi un}{N} \right] \quad (2.11)$$

para  $u = 0, 1, 2, \dots, N-1$  e a transformada inversa de Fourier de  $F(u)$  será:

$$f(n) = \frac{1}{N} \sum_{u=0}^{N-1} F(u) \exp \left[ \frac{-2j\pi un}{N} \right] \quad (2.12)$$

para  $n = 0, 1, 2, \dots, N-1$ .

Os valores  $u = 0, 1, 2, \dots, N-1$  na transformada discreta de Fourier (Equação 2.11) correspondem a amostras dos valores da transformada do sinal contínuo nos pontos  $0, \Delta u, 2\Delta u, \dots, (N-1)\Delta u$ . Em outras palavras,  $F(u)$  representa  $F(u\Delta u)$ . Portanto, os intervalos de espaçamento entre as amostras do sinal e de sua transformada estão relacionados através da expressão

$$\Delta u = \frac{1}{N\Delta x} \quad (2.13)$$

### 2.5.2 Transformada discreta de Fourier (2-D)

Seja agora uma função contínua bidimensional  $f(x, y)$ , discretizada em  $M$  amostras ao longo de  $x$  e  $N$  amostras ao longo de  $y$ . Neste caso, sua transformada discreta de Fourier será:

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \exp \left[ -j2\pi \left( \frac{ux}{M} + \frac{vy}{N} \right) \right] \quad (2.14)$$

para  $u = 0, 1, 2, \dots, M-1, v = 0, 1, 2, \dots, N-1$ . A transformada inversa é dada por:

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) \exp \left[ j2\pi \left( \frac{ux}{M} + \frac{vy}{N} \right) \right] \quad (2.15)$$

O processo de amostragem da função bidimensional contínua  $f(x, y)$  pode ser visto como uma grade 2-D, com impulsos unitários espaçados de  $F_x$  e  $F_y$ , nos eixos  $x$  e  $y$  respectivamente. A função  $f(x, y)$  discretizada (utilizada nas Equações 2.14 e 2.15) representa as amostras de  $f(x, y)$  contínua original em pontos espaçados entre si de  $F_x$  e  $F_y$ , nos respectivos eixos. Os intervalos de amostragem nos domínios espacial de frequência estão relacionados entre si por:

$$\Delta u = \frac{1}{M\Delta x} \quad (2.16)$$

e

$$\Delta v = \frac{1}{N\Delta y} \quad (2.17)$$

Para o caso particular em que  $M = N$ , as equações (2.13) e (2.14) podem ser reescritas como:

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \exp \left[ -j2\pi \left( \frac{ux + vy}{N} \right) \right] \quad (2.18)$$

e

$$f(x, y) = \frac{1}{N} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) \exp \left[ j2\pi \left( \frac{ux + vy}{N} \right) \right] \quad (2.19)$$

Notar que o termo  $1/MN$  foi desmembrado em dois termos  $1/N$ , um em cada equação, uma vez que o agrupamento ou desmembramento destas constantes multiplicativas é arbitrário.

Neste trabalho, a transformada de Fourier foi necessária para a transformação da imagem para o domínio de Fourier, pois nesse domínio, as operações necessárias para gerar as matrizes circulantes se tornam mais simples e menos custosas.

## 2.6 MODELO DA REGRESSÃO LINEAR

A regressão nasce da tentativa de relacionar um conjunto de observações de certas variáveis, designadas genericamente por  $X_k (k = 1..p)$ , com as leituras de uma certa grandeza  $Y$ . No caso da regressão linear, está subjacente uma relação do tipo:

$$Y = a + b_1X_1 + b_2X_2 + \dots + b_pX_p \quad (2.20)$$

onde  $a, b_1, b_2, \dots, b_p$  são os parâmetros da relação linear procurada. O objetivo pode ser explicativo (demonstrar uma relação matemática que pode indicar, mas não prova,

uma relação de causa-efeito) ou preditivo (obter uma relação que nos permita, perante futuras observações das variáveis  $X_k$ , prever o correspondente valor de  $Y$ , sem necessidade de o medir). Dadas as características deste texto, não se aprofundará esta questão, mas a distinção básica entre as duas situações é fundamental. Independentemente dos objetivos, as variáveis  $X_k$  são muitas vezes designadas por variáveis explicativas, uma vez que tentam explicar as razões da variação de  $Y$ .

Supondo que se dispõe de  $n$  conjuntos de medidas com as correspondentes observações, a utilização do modelo incluirá sempre uma parcela de erro. Utilizando o índice  $i$  ( $i = 1..n$ ) para indicar cada conjunto, ter-se-á então:

$$y_i = a + b_1x_{i1} + b_2x_{i2} + \dots + b_px_{ip} + e_i \quad i = 1..n \quad (2.21)$$

Independentemente das motivações, a versão da regressão linear que aqui se apresenta consiste em estimar os valores dos parâmetros  $a, b_1, b_2, \dots, b_p$ , através da minimização da soma dos quadrados dos desvios. Daí o nome de método dos mínimos quadrados que às vezes se utiliza, nomeadamente para a regressão simples ( $p = 1$ ). O termo multi-regressão é usado para explicitar o caso  $p > 1$ . Neste ponto, é conveniente definir:

- $y = [y_1y_2..y_n]'$  vetor das leituras;
- $x_k = [x_{1k}x_{2k}..x_{nk}]'$  vetor das observações de cada variável  $X_k$ ;
- $X = [x_1x_2..x_p]$  matriz de observações (elementos  $x_{ik}$ ,  $i=1..n$ ,  $k=1..p$ );
- $b = [ab_0]' = [ab_1b_2..b_p]'$  vetor dos parâmetros;
- $e = [e_1e_2..e_n]'$  vetor dos erros;
- $1 = [1..1]'$  vetor unitário de dimensão  $n$ ;
- $Xa = [1x_1x_2..x_p]$  matriz aumentada de observações ( $i=0..n$ ,  $k=0..p$ ).

Com estas definições, é possível escrever a expressão anterior de forma simplificada:

$$y = a.1 + X.b_0 + e \quad (2.22)$$

ou

$$y = X_a.b + e \quad (2.23)$$

Uma vez obtida a estimativa  $\hat{b}$  dos parâmetros  $b$ , a expressão operacional da regressão permite obter estimativas  $\hat{y}$  das leituras correspondentes às observações  $x_1 x_2 \dots x_p$ :

$$\hat{y} = \hat{a} + \hat{b}_1 x_1 + \hat{b}_2 x_2 + \dots + \hat{b}_p x_p \quad (2.24)$$

ou

$$\hat{y} = \hat{a} + \hat{x}' \cdot \hat{b}_0 \quad (2.25)$$

O cálculo simultâneo dos valores estimados correspondentes às observações usadas na parametrização da regressão pode ser feito com base na expressão matricial correspondente:

$$\hat{y} = X_x \cdot \hat{b} \quad (2.26)$$

## 2.7 DESLOCAMENTOS CÍCLICOS

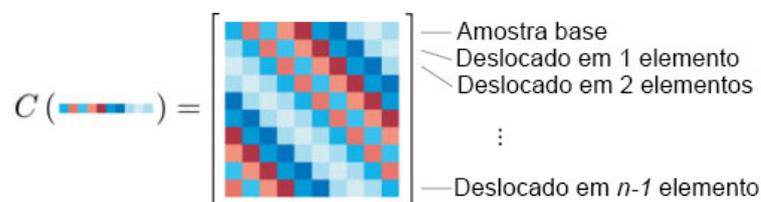
Considerando-se um vetor  $n \times 1$  representando uma amostra com o objeto de interesse, denotado por  $x$ , referindo-se a ele como a amostra base. O objetivo é treinar um classificador com a amostra base (um exemplo positivo) e várias amostras virtuais obtidas por translação (que servem como exemplos negativos) (HENRIQUES et al., 2014). Pode-se modelar translações unidimensionais desse vetor por um operador de deslocamento cíclico, que é a matriz de permutação

$$P = \begin{bmatrix} 0 & 0 & 0 & \dots & 1 \\ 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix} \quad (2.27)$$

O produto  $Px = [x_1, x_2, \dots, x_{n-1}]^T$  desloca  $x$  por um elemento, modelando uma pequena translação. Podemos encadear  $u$  deslocamentos para obter uma translação maior usando a matriz  $P^u x$ . Um  $u$  negativo realizará o mesmo processo na direção inversa.

A Figura 8 mostra a ilustração de uma matriz circulante. As linhas são deslocamentos cíclicos de uma imagem vetorial ou suas translações em 1D. As mesmas propriedades são transferidas para matrizes circulantes de imagens 2D.

FIGURA 8 – Deslocamento cíclicos 1d.



Fonte: Adaptado de Henriques et al. (2014)

A Figura 9 mostra exemplos de deslocamentos cíclicos verticais de uma amostra de base. A formulação de domínio de Fourier permite treinar um rastreador com todos os deslocamentos cíclicos possíveis de uma amostra de base, tanto vertical quanto horizontal, sem iterá-los explicitamente. Artefatos das bordas envolvidas podem ser vistos (parte superior da imagem mais à esquerda), mas são atenuados pela janela do co-seno e pelo preenchimento.

FIGURA 9 – Deslocamento cíclicos 2d.



Fonte: Adaptado de Henriques et al. (2014)

Devido à propriedade cíclica, obtém-se o mesmo sinal  $x$  periodicamente a cada  $n$  turnos. Isto significa que o conjunto completo de sinais deslocados é obtido com

$$\{p^u x | u = 0, \dots, n - 1\} \quad (2.28)$$

Novamente devido à propriedade cíclica, é possível ver equivalentemente a primeira metade deste conjunto como desvios na direção positiva, e a segunda metade como desvios na direção negativa.

A técnica de deslocamento cíclico foi usada neste trabalho para gerar matrizes circulantes.

## 2.8 MATRIZES CIRCULANTES

Para calcular uma regressão com amostras deslocadas (HENRIQUES et al., 2014), podemos usar o conjunto de Equação (2.28) como as linhas de uma matriz de

dados  $X$ :

$$X = C(x) = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_n \\ x_n & x_1 & x_2 & \cdots & x_{n-1} \\ x_{n-1} & x_n & x_1 & \cdots & x_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_2 & x_3 & x_4 & \cdots & x_1 \end{bmatrix} \quad (2.29)$$

Uma ilustração do padrão resultante é dada na Figura 9. O que se obtém como resultado é uma matriz circulante, que tem várias propriedades importantes. Observa-se que o padrão é determinístico e totalmente especificado pelo vetor gerador  $x$ , que é a primeira linha.

Outro importante e útil fato é que todas as matrizes circulantes são tornadas diagonais pela DFT, independentemente do vetor gerador  $x$ . Isso pode ser expresso como

$$X = F \text{diag}(\hat{x}) F^H \quad (2.30)$$

onde  $F$  é uma matriz constante que não depende de  $x$ , e  $\hat{x}$  denota a DFT do vetor gerador,  $\hat{x} = F(x)$ .

A matriz constante  $F$  é conhecida como a matriz DFT e é a matriz exclusiva que calcula a DFT de qualquer vetor de entrada, como  $F(z) = \sqrt{n}Fz$ . Isso é possível porque a DFT é uma operação linear.

Equação (2.30) expressa a decomposição espectral de uma matriz circulante geral. Os autovetores  $F$  determinísticos compartilhados estão na raiz de muitas características incomuns, como a comutatividade ou a inversão de forma fechada.

Neste trabalho, como um caso especial de modelo de regressão (apresentado na seção 2.5), os filtros de correlação geram seu conjunto de treinamento deslocando ciclicamente uma amostra de base.

## 2.9 FILTRO DE CORRELAÇÃO

Um filtro de correlação é um modelo projetado especificamente para reconhecer uma determinada classe de padrões representada por um conjunto de padrões de referência. Um determinado padrão de consulta é comparado com esse modelo executando uma correlação cruzada. Para tornar isso eficiente, a correlação cruzada é realizada no domínio da frequência, aproveitando a Transformada Rápida de Fourier

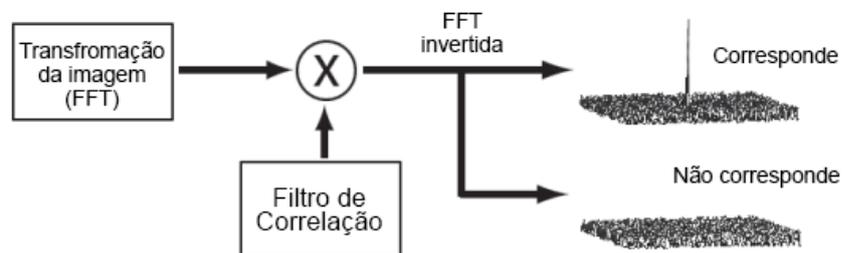
(FFT).

$$C(x, y) = Ft^{-1}FTI(x, y) \cdot F^*(uv) \quad (2.31)$$

onde  $I(x, y)$  é um padrão de consulta e  $F(u, v)$  é a representação do domínio da frequência do filtro de correlação.

A saída de correlação cruzada resultante  $C(x, y)$  deve conter um pico agudo se a consulta for autêntica e nenhum pico, se for falsa, como mostrado na figura 10.

FIGURA 10 – Filtro de correlação e sua saída



Fonte: Adaptado de Boddeti e Kumar (2008)

As principais vantagens de usar filtros de correlação são:

1. Geração de todo o plano de correlação de uma única vez.
2. Performance razoável de desempenho mesmo na presença de ruído ou oclusões.
3. Pode ser projetado para tolerar uma variedade de variações dentro da classe.

Para quantificar o grau de nitidez do pico de correlação, usa-se a razão entre pico e correlação (PCE) definida como:

$$PCE = \frac{peak - \mu}{\sigma} \quad (2.32)$$

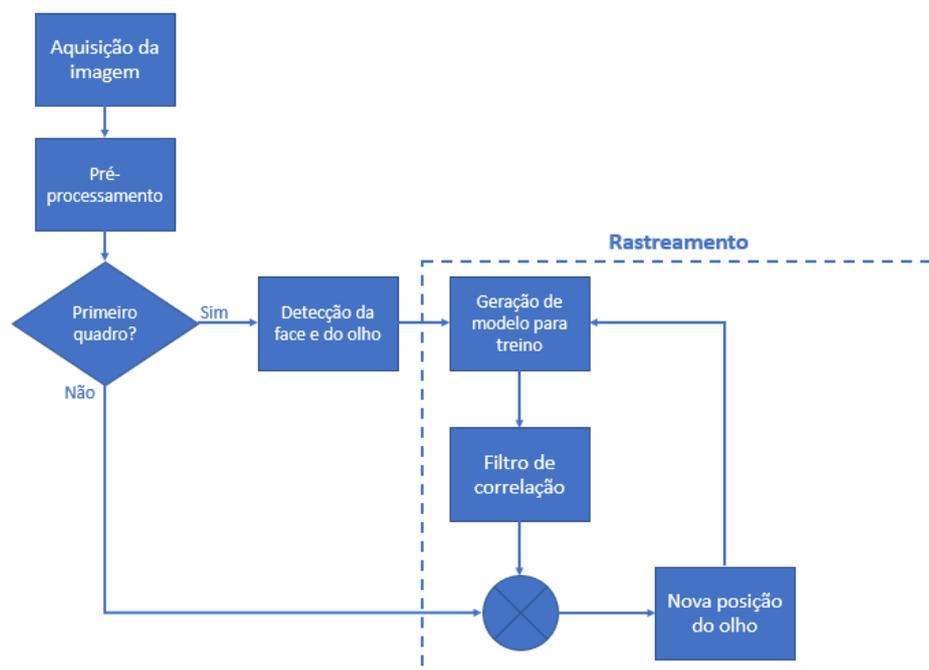
onde  $m$  e  $s$  são a média e o desvio padrão do plano de correlação, respectivamente.

Neste trabalho, o filtro de correlação é empregado na fase de rastreamento do olho, na identificação do quadro que tem o maior pico de correlação, e, conseqüentemente, o quadro que possivelmente contem o objeto que está sendo rastreado.

### 3 MÉTODO PROPOSTO

Este capítulo apresenta o método proposto para realizar o rastreamento dos olhos. Para isso, de acordo com os objetivos específicos definidos no capítulo 1, a metodologia foi dividida segundo o seguinte esquema (Figura 11):

FIGURA 11 – Esquema do método proposto



Fonte: Acervo do autor

Abaixo, segue uma breve explicação de cada uma das etapas da metodologia.

#### a) Aquisição da imagem

Os vídeos utilizados neste trabalho foram capturados utilizando *webcam* e fazendo uso de funções de captura de vídeo presente na biblioteca OpenCV.

#### b) Pré-processamento

Nesta etapa realiza-se o pré-processamento da imagem, que consiste na conversão da imagem em escala de cinza e a equalização do histograma, como descrito na Seção 2.1. Após preparar o quadro, é verificado se o rastreador foi ou não inicializado. Se o rastreador já estiver sido inicializado e o olho detectado, então o algoritmo segue para a etapa de rastreamento. Caso contrário, segue para etapa de detecção da face e do olhos.

#### c) Detecção da face e do olho

A captura do olho é feita automaticamente. Essa etapa utiliza o classificador

em cascata para os quadros seguintes, incluindo o atual, até que o olho seja detectado. Uma vez que a detecção tenha ocorrido com sucesso, a sub-imagem que contém o olho é capturada e extraída para ser usada na etapa seguinte.

#### d) Rastreamento do olho

A sub-imagem obtida na etapa anterior é utilizada para o treinamento do filtro de correlação. Após o treinamento ter sido realizado, captura-se um novo quadro, e delimita-se uma região de busca a partir da coordenada do olho do capturado no quadro anterior. Esta região terá um tamanho 2,5 vezes maior que o quadro do olho. Então, realiza-se uma busca em toda essa região com o filtro de correlação que foi treinado. O local desta região que retornar o maior pico será eleito como novo local do olho. Após a definição da nova localização do olho, uma nova sub-imagem do olho é extraída e usada para treinar o filtro de correlação. Finalmente, repetem-se as etapas.

A seguir, serão apresentadas mais detalhadamente cada uma das etapas da metodologia, evidenciando-se detalhes da implementação.

### 3.1 PRÉ-PROCESSAMENTO

Para se realizar o rastreamento do olho, é necessário capturar a imagem do usuário, transformar esta imagem para a escala de cinza, e então é feita uma melhoria através da equalização de histograma.

O pré-processamento do vídeo é uma etapa importante para o funcionamento da metodologia. Primeiramente, porque todas as etapas seguintes dependem dos quadros extraídos do vídeo. Além disso, a qualidade das imagens extraídas dos quadros afetam diretamente as técnicas de processamento de imagens utilizadas.

Há diversos fatores na aquisição do vídeo que podem influenciar nas etapas posteriores, entre eles a resolução de cada quadro. Imagens de alta resolução, por si só, não são o suficiente para obter-se melhores resultados. Outros fatores também afetam diretamente as técnicas aqui empregadas, tais como a presença de ruído e a iluminação instável.

Outro fator a ser destacado é que o método computacional está fortemente voltado para computadores convencionais. Por isso, os quadros capturados serão, quase sempre, obtidos a partir de webcams. Isso significa que, na prática, utiliza-se quadros com resolução mediana e, provavelmente, de qualidade mediana.

O pré-processamento realizado durante a aquisição, conforme o diagrama da Figura 13, trata-se da obtenção do quadro, a conversão em escala de cinza e subsequente equalização do histograma. A equalização do histograma da imagem é realizada para melhorar possíveis imagens com um baixo contraste.

### 3.2 DETECÇÃO DA FACE E DO OLHO

Para realizar a detecção de olho, inicialmente realiza-se a detecção de face sobre o quadro. Utilizando-se o classificador em cascata, é realizada a tentativa de detecção. Caso uma face não seja detectada, o mesmo procedimento é repetido para o quadro seguinte até obter-se um resultado. Somente após uma detecção da face bem sucedida, é extraída a região da face e nela é aplicada a etapa de detecção dos olhos. Caso não seja possível detectar um dos olhos então o método retorna para a detecção de face.

Ambas as etapas são aplicadas a partir do método de detecção de objetos do classificador em cascata baseado nas *Haar-Like Features* apresentado na Seção 2.3. A biblioteca OpenCV dá suporte para ambas as detecções.

Depois da detecção da região do olho, emprega-se a transformada de Hough para regiões circulares como descrito na Seção 2.2 para detectar o contorno da íris e assim estimar-se o centro do olho. Uma vez que a detecção tenha sido realizada com sucesso, uma *bounding box* do olho é obtida a partir do centro do olho. Dessa *bounding box*, será extraída uma sub-imagem que será utilizada para o treinamento do filtro de correlação na próxima etapa.

Alguns pontos devem ser levados em consideração nessa etapa:

- Uma vez que o classificador em cascata consiga identificar o olho com sucesso, a sub-imagem resultante desse processo será utilizada para treinamento do filtro de correlação;
- Essa etapa é ignorada caso o olho esteja sendo rastreado. Como descrito anteriormente, essa etapa busca extrair a sub-imagem a ser utilizada para inicializar o rastreamento.

O esquema da Figura 12 mostra o funcionamento dessa etapa:

FIGURA 12 – Esquema da fase de detecção da face e dos olhos



Fonte: Acervo do autor

### 3.3 RASTREAMENTO DO OLHO

Rastreadores baseados em filtro modelam a aparência de objetos usando filtros treinados em imagens de exemplo. E isso não muda com o filtro de correlação. Depois da etapa de detecção do olho, a *bounding box* contendo a imagem do olho é definida como o alvo do rastreamento.

A partir deste ponto, o rastreamento e o treinamento do filtro funcionam em conjunto. O alvo é rastreado correlacionando o filtro em uma janela de pesquisa no próximo alvo. Esta janela é 2,5 vezes maior que o alvo. A localização correspondente ao valor máximo na saída do filtro de correlação indica a nova posição do alvo e uma atualização em tempo real é executada com base nesse novo local.

Para tornar o rastreador rápido, a correlação é calculada no domínio de Fourier (Seção 2.5). Primeiro, a transformada discreta de Fourier 2D da imagem de entrada e do filtro são calculados. O Teorema da Convolução afirma que a correlação se torna uma multiplicação elementar no domínio de Fourier. (BOLME; BEVERIDGE et al., 2010) As saída da correlação é transformada de volta ao domínio espacial usando a FFT inversa.

O custo nesse processo é calcular as FFTs direta e inversa, de modo que todo o processo tenha um tempo limite superior de  $O(P \log P)$ , onde  $P$  é o número de *pixels* na janela de rastreamento. O conjunto de treinamento é feito por meio de um modelo de regressão, onde o filtro de correlação, usa a amostra base (sub-imagem do olho) como exemplo positivo, e várias imagens transladadas obtidas através do deslocamento cíclico e das matrizes circulantes (apresentadas nas seções 2.7 e 2.8) como exemplos negativos.

As saídas de treinamento também são geradas com seus picos correspondentes ao centro do alvo. Durante o rastreamento, um alvo pode mudar a aparência alterando a rotação, escala, posicionamento, movendo-se através de diferentes condições de iluminação ou até mesmo passando por uma deformação. Portanto, os filtros precisam se adaptar rapidamente para seguir os objetos.

### 3.3.1 Algoritmo e parametrização do filtro

O algoritmo usado para a realização do rastreamento é demonstrado abaixo na linguagem Matlab<sup>1</sup>:

```

\% Entradas
\% x: sub-imagem para treino (m x n x c)
\% y: alvo da regressao, com kernel gaussiano (m x n)
\% z: sub-imagem de teste (m x n x c)

\% Saida
\% retorno: score de deteccao para cada local, m x n

function alphaf = treinar (x, y, sigma, lambda)
    k = correlacao (x, x, sigma);
    alphaf = fft2(y) ./ (fft2(k) + lambda);
end

function retorno = detectar (alphaf, x, z, sigma)
    k = correlacao (z, x, sigma);
    retorno = real(ifft2(alphaf .* fft2(k)));
end

function k = correlacao (x1, x2, sigma)
    c = ifft2(sum(conj(fft2(x1)) .* fft2(x2), 3));
    d = x1(:)' * x1(:) + x2(:)' * x2(:) - 2 * c;
    k = exp(-1 / sigma^2 * abs(d) / numel(d));
end

```

Onde:

- O  $\lambda$  é um parâmetro de regularização que controla o overfitting<sup>2</sup>
- O  $\sigma$  é o desvio padrão da distribuição gaussiana

<sup>1</sup> MATLAB é uma abreviatura de "laboratório de matriz". Enquanto outras linguagens de programação geralmente trabalham com números um de cada vez, o MATLAB® opera em matrizes e matrizes inteiras. Os fundamentos da linguagem incluem operações básicas, como a criação de variáveis, indexação de arrays, aritmética e tipos de dados. Disponível em: <https://www.mathworks.com/help/matlab/language-fundamentals.html>

<sup>2</sup> Sobre-ajuste ou sobreajuste (do inglês: *overfitting*) é um termo usado em estatística para descrever quando um modelo estatístico se ajusta muito bem ao conjunto de dados anteriormente observado, mas se mostra ineficaz para prever novos resultados.

No primeiro quadro, um modelo é treinado com a sub-imagem do olho na posição inicial. Este quadro tem uma região maior que o alvo, para fornecer um contexto ao rastreador. Para cada novo quadro, detectamos sobre o região na posição anterior e a posição de destino é atualizada para a que gerou a saída de maior pico. Finalmente, é treinado um novo modelo com a nova posição, e os valores de  $\alpha$  e  $x$  são interpolados linearmente com os do quadro anterior, para fornecer o rastreador com alguma memória.

Neste trabalho, depois de alguns testes, os parâmetros com os quais se obteve o melhor resultado foram  $\sigma = 0.05$  e  $\lambda = 0.0001$ .

## 4 TESTES E RESULTADOS

Durante o desenvolvimento desse trabalho foi implementada uma aplicação a fim de testar a metodologia aqui proposta, fazendo, assim, uma análise e uma avaliação da metodologia. Foi utilizada uma *webcam* de 2.0 megapixels em um computador com processador Intel Core i5 de 2<sup>o</sup> Geração, com 8 *gigabytes* de RAM .

A seguir serão apresentadas as métricas usadas para a avaliação deste trabalho, bem como os testes realizados nas etapas da detecção e do rastreamento, individualmente.

### 4.1 MÉTRICAS

O termo “métricas” refere-se a estatísticas gerais e medidas de desempenho. Ele pode ser entendido como um meio de mensuração de desempenho por meio de uma coleção de dados específicos ou gerais. Para a obtenção das métricas, os resultados são divididos nas seguintes classes:

- Verdadeiros Positivos (VP)
- Falsos positivos (FP)
- Verdadeiro Negativo (VN)
- Falsos Negativos (FN)

A partir desses valores, é possível realizar o cálculo de métricas relevantes para o método proposto. A seguir, apresenta-se as métricas utilizadas para avaliar a aplicação do método.

- Acurácia

A proporção de predições corretas, sem levar em consideração o que é positivo e o que é negativo. Esta medida é altamente suscetível a desbalanceamentos do conjunto de dados e pode facilmente induzir a uma conclusão equivocada sobre o desempenho do sistema.

Acurácia = Total de acerto / Total de dados no conjunto

$$Acuracia = \frac{(VP+VN)}{(TOTAL)}$$

- Precisão

A proporção de verdadeiros em relação a todos os positivos: A proporção de verdadeiros positivos em relação a todas as predições positivas.

Precisão = Acertos Positivos / Total de predições positivas

$$Precisao = \frac{VP}{(VP+FP)}$$

- Sensibilidade

A proporção de verdadeiros positivos: a capacidade do método em prever corretamente a condição para casos que realmente a têm.

Sensibilidade = Acertos positivos / Total de positivos

$$Sensibilidade = \frac{VP}{(VP+FN)}$$

- Especificidade

A proporção de verdadeiros negativos: a capacidade do método em prever corretamente a ausência da condição para casos que realmente não a têm.

Especificidade = Acertos negativos / Total de negativos

$$Especificidade = \frac{VN}{(VN+FP)}$$

- F1-Score

A média aritmética da Sensibilidade e Especificidade. Na prática, a sensibilidade e a especificidade variam em direções opostas. Isto é, geralmente, quando um método é muito sensível a positivos, tende a gerar muitos falso-positivos, e vice-versa. Assim, um método de decisão perfeito (100% de sensibilidade e 100% especificidade) raramente é alcançado, e um balanço entre ambos deve ser atingido.

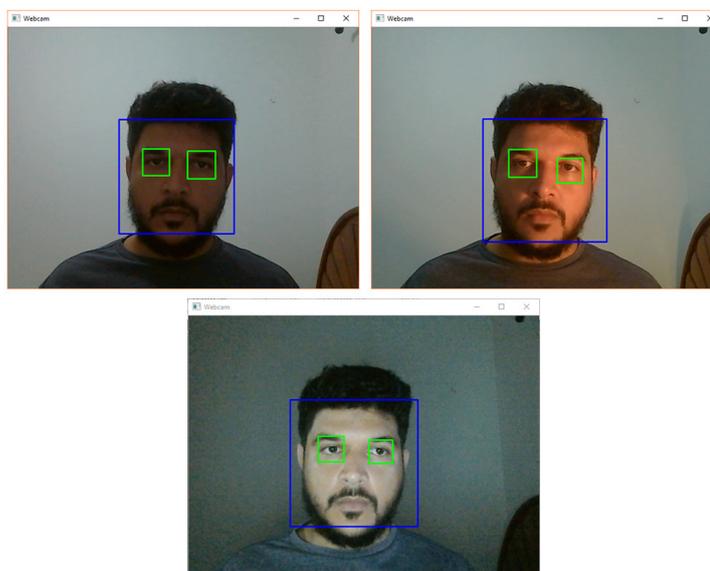
$$F1 - Score = \frac{(Sensibilidade+Especificidade)}{2}$$

## 4.2 TESTES DE DETECÇÃO DO OLHO

A captura do olho realizada faz uso de um classificador em cascata que utiliza as características *Haar-like*. A Figura 13 mostra os resultados dessa etapa, onde a detecção de olho foi precedida da detecção de face.

Na fase de detecção, o algoritmo retorna um vetor com todos os olhos encontrados na imagem. Testou-se o rastreamento em um dos olhos. Entretanto o método pode ser ampliado para rastrear os dois olhos.

FIGURA 13 – Detecção dos olhos



Fonte: Acervo do autor

Foram realizados testes de detecção dos olhos em 50 imagens, obtidas pela *webcam*, em várias condições de iluminação e em que a face está em diferentes ângulos. Dessas 50, 15 não apresentavam olhos. A Figura 14 apresenta os resultados da detecção dos olhos.

FIGURA 14 – Resultado da detecção dos olhos

Métricas	Valor
Número de imagens	50
Falso positivos	4
Falso negativo	9
Verdadeiro positivo	26
Verdadeiro negativo	11
<b>Acurácia</b>	0,74
<b>Precisão</b>	0,87
<b>Sensibilidade</b>	0,74
<b>Especificidade</b>	0,73
<b>Eficiência</b>	0,74

Fonte: Acervo do autor

#### 4.3 RESULTADOS DO RASTREAMENTO DO OLHO

Como proposto na Seção 3.3, o rastreamento do olho se deu através de um filtro de correlação. (Figura 15)

Para se avaliar o rastreador, foi necessário a elaboração de um método de teste que permitisse extrair métricas para a avaliação. Para isso, realizou-se um teste

FIGURA 15 – Rastreamento do olho



A imagem apresenta 4 tipos de iluminação: a) iluminação ambiente; b) iluminação ambiente e luz direta; c) luz direta sem iluminação ambiente; e d) fonte de luz direta fraca. Fonte: Acervo do autor

em 80 vídeos de 3 segundos, obtidos através de uma *webcam* (Figura 15). Os 80 vídeo foram divididos em 4 grupos de 20 vídeos, onde cada grupo possui um nível diferente iluminação (Figura 15). Além disso, em cada nível de iluminação foram gravados 5 vídeos com alterações de escala (Figura 17), e 5 vídeos com alterações de rotação (Figura 18). Os pontos inicial e final do olho em cada vídeo foram marcados manualmente, com o objetivo de verificar se os pontos, inicial e final, do rastreador coincidem com os pontos marcados, com uma tolerância de 10 *pixels* (Figura 16).

FIGURA 16 – Teste do rastreamento do olho



Fonte: Acervo do autor

Em cada vídeo, verificou-se que o método proposto rastreou e os resultado dessa verificação foi modelada da seguinte forma:

- Verdadeiro Positivo (VP): O rastreador obteve sucesso em rastrear o olho quando

FIGURA 17 – Teste do rastreamento do olho com variação de escala



Fonte: Acervo do autor

FIGURA 18 – Teste do rastreamento do olho com variação de rotação



Fonte: Acervo do autor

havia olho no vídeo.

- Verdadeiro Negativo (VN): O rastreador não rastreou o olho, e não havia olho no vídeo.
- Falso Positivo (FP): O rastreador não rastreou o olho, e rastreou outra parte da imagem, mesmo havendo olho no vídeo.
- Falso Negativo (FN): O rastreador não rastreou nada, mesmo havendo olho no vídeo.

#### 4.3.1 Testes e resultados

Os testes foram realizados em uma base de de 80 vídeos (Figura 19) criada para o método proposto neste estudo.

Pelo resultado apresentado na Figura 19, é possível notar um bom nível de acerto e um bom nível de precisão. Além disso, a sensibilidade demonstra que o método tem um boa taxa de acertos positivos, ou seja, o olho é rastreado corretamente na maioria das vezes.

Os resultados seguintes, foram realizados com sub-conjuntos do resultado geral.

No primeiro sub-conjunto, foi avaliado o quanto a alteração de escala e a rotação interferem no rastreamento. O resultado é apresentado na Figura 20.

FIGURA 19 – Resultado do rastreamento

Métrica	Valor
Número de vídeos	80
VP	48
VN	10
FP	12
FN	10
<b>Acurácia</b>	0,73
<b>Precisão</b>	0,80
<b>Sensibilidade</b>	0,83
<b>Especificidade</b>	0,45
<b>Eficiência</b>	0,64

Fonte: Acervo do autor

FIGURA 20 – Resultado dos sub-conjuntos que possuem variação de rotação e escala

Rotação		Escala	
Métrica	Valor	Métrica	Valor
Número de vídeos	20	Número de vídeos	20
VP	9	VP	11
VN	4	VN	1
FP	5	FP	4
FN	2	FN	4
<b>Acurácia</b>	0,65	<b>Acurácia</b>	0,60
<b>Precisão</b>	0,64	<b>Precisão</b>	0,73
<b>Sensibilidade</b>	0,82	<b>Sensibilidade</b>	0,92
<b>Especificidade</b>	0,44	<b>Especificidade</b>	0,20
<b>Eficiência</b>	0,63	<b>Eficiência</b>	0,56

Fonte: Acervo do autor

Nos resultados apresentados na Figura 20, é possível observar a influência da rotação e escala no resultado do rastreamento. A acurácia e precisão tiveram queda significativa, o que nos indica que o rastreador tem dificuldade em seguir o alvo durante a rotação e mudança de escala. Além disso a especificidade também teve queda, o que indica que o rastreador tem mais dificuldade em identificar que não há olho na imagem quando a escala é alterada.

Os próximos resultados são relacionados ao nível de iluminação do vídeo. Testes foram realizados com 4 níveis diferentes de iluminação, em um total de 20 vídeos para cada tipo de iluminação. A Figura 21 apresenta os resultados.

Pela Figura 21 é possível observar que o rastreamento no vídeo com maior iluminação tem um performance melhor que os demais. Isso indica que ainda que o método tenha um pré-processamento na tentativa de evitar problemas de iluminação, o

FIGURA 21 – Resultado dos sub-conjuntos que possuem variação iluminação

a		b	
Métrica	Valor	Métrica	Valor
Número de vídeos	20	Número de vídeos	20
VP	16	VP	13
VN	0	VN	3
FP	2	FP	3
FN	2	FN	1
<b>Acurácia</b>	0,80	<b>Acurácia</b>	0,80
<b>Precisão</b>	0,89	<b>Precisão</b>	0,81
<b>Sensibilidade</b>	0,89	<b>Sensibilidade</b>	0,93
<b>Especificidade</b>	0,00	<b>Especificidade</b>	0,50
<b>Eficiência</b>	0,44	<b>Eficiência</b>	0,71

c		d	
Métrica	Valor	Métrica	Valor
Número de vídeos	20	Número de vídeos	20
VP	9	VP	9
VN	3	VN	3
FP	6	FP	6
FN	2	FN	2
<b>Acurácia</b>	0,60	<b>Acurácia</b>	0,60
<b>Precisão</b>	0,60	<b>Precisão</b>	0,60
<b>Sensibilidade</b>	0,82	<b>Sensibilidade</b>	0,82
<b>Especificidade</b>	0,33	<b>Especificidade</b>	0,33
<b>Eficiência</b>	0,58	<b>Eficiência</b>	0,58

As tabelas a, b, c, d, correspondem a níveis de iluminação iguais aos da Figura 15. Fonte: Acervo do autor

modo como a imagem é adquirida pode interferir muito no resultado do rastreamento.

## 5 CONCLUSÃO

Esse trabalho teve por objetivo desenvolver um método computacional para rastreamento do olho em vídeo utilizando um filtro de correlação. A metodologia tem como base quatro técnicas: A detecção da face; a detecção do olho; o treinamento do filtro de correlação; e o rastreamento do olho com base no filtro de correlação. Os resultados endossam a eficácia do método.

O trabalho apresenta uma contribuição na área de rastreamento de olhos, pois demonstra que o uso de rastreadores baseados em filtro de correlação podem ser adaptados para a detecção do olho. Os resultados obtidos mostram que o método proposto é capaz de realizar o rastreamento do olho de forma eficiente e satisfatória, podendo ser utilizado em diversas aplicações, como controle do ponteiro do mouse com os olhos ou no auxílio em exames médicos.

O método proposto apresentou bons resultados, como discutido no capítulo anterior. Esse trabalho apresenta uma metodologia flexível, que permite modificações que possam aprimorar a técnica de rastreamento.

Este estudo avaliou o uso de um método de rastreamento pouco utilizado para o fim de rastreamento dos olhos. Os métodos de rastreamento baseados em filtro de correlação, como já é amplamente utilizado para rastrear outros tipos de objetos, também podem se mostrar eficientes quando o alvo do rastreamento é o olho.

Visando-se trabalhos futuros, são sugeridas algumas modificações que podem melhorar os resultados:

1. Usar mais de uma imagem como treinamento do caso positivo;
2. Oferecer mais informação de contexto ao rastreador;
3. Testar outras técnicas de pré-processamento a fim de melhorar os resultados em diferentes níveis de iluminação.

## REFERÊNCIAS

- ADAM, A.; RIVLIN, E.; SHIMSHONI, I. Robust Fragments-based Tracking using the Integral Histogram. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06). [S.l.: s.n.], jun. 2006. v. 1, p. 798–805. DOI: 10.1109/CVPR.2006.256. Citado 1 vez na página 12.
- ASTRONOMY, COSMOS - The SAO Encyclopedia of. **Gaussian Function**. [S.l.: s.n.], 2008. <https://astronomy.swin.edu.au/cosmos/G/Gaussian+Function>. Acessado: 10 maio 2019. Citado 1 vez na página 23.
- BODDETI, Naresh; KUMAR, BVK Vijaya. Extended depth of field iris recognition with correlation filters. In: IEEE. 2008 IEEE Second International Conference on Biometrics: Theory, Applications and Systems. [S.l.: s.n.], 2008. p. 1–8. Citado 0 vez na página 30.
- BOLME, D. S.; BEVERIDGE, J. R. et al. Visual object tracking using adaptive correlation filters. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. [S.l.: s.n.], jun. 2010. p. 2544–2550. DOI: 10.1109/CVPR.2010.5539960. Citado 2 vezes nas páginas 14, 34.
- BOLME, D. S.; DRAPER, B. A.; BEVERIDGE, J. R. Average of Synthetic Exact Filters. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. [S.l.: s.n.], jun. 2009. p. 2105–2112. DOI: 10.1109/CVPR.2009.5206701. Citado 1 vez na página 14.
- BRADSKI, Gary; KAEHLER, Adrian. **Learning OpenCV: Computer Vision in C++ with the OpenCV Library**. 2nd. [S.l.]: O'Reilly Media, Inc., 2013. Citado 2 vezes nas páginas 18, 20.
- DANELLJAN, Martin et al. Accurate scale estimation for robust visual tracking. In: BMVA PRESS. BRITISH Machine Vision Conference, Nottingham, September 1-5, 2014. [S.l.: s.n.], 2014. Citado 1 vez na página 14.
- GIANNOTTO, Eduardo Chimati. **Uso de rastreamento do olhar na avaliação da experiência do tele-usuário de aplicações de TV interativa**. 2009. Mestrado em Sistemas Digitais – Escola Politécnica, Universidade de São Paulo. Citado 1 vez na página 12.
- HARE, Sam et al. Struck: Structured output tracking with kernels. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 38, n. 10, p. 2096–2109, 2015. Citado 1 vez na página 14.
- HENRIQUES, João F. et al. High-Speed Tracking with Kernelized Correlation Filters. **CoRR**, abs/1404.7584, 2014. arXiv: 1404.7584. Disponível em: <<http://arxiv.org/abs/1404.7584>>. Citado 7 vezes nas páginas 13, 14, 27, 28.

KALAL, Zdenek; MIKOLAJCZYK, Krystian; MATAS, Jiri. Tracking-learning-detection. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 34, n. 7, p. 1409–1422, 2011. Citado 1 vez na página 14.

MAHALANOBIS, Abhijit et al. Unconstrained correlation filters. **Applied Optics**, Optical Society of America, v. 33, n. 17, p. 3751–3759, 1994. Citado 1 vez na página 14.

MARQUES FILHO, Ogê; NETO, Hugo Vieira. **Processamento digital de imagens**. [S.l.]: Brasport, 1999. Citado 2 vezes nas páginas 16–18, 23.

ROSS, David A. et al. Incremental Learning for Robust Visual Tracking. **International Journal of Computer Vision**, v. 77, n. 1, p. 125–141, maio 2008. ISSN 1573-1405. DOI: 10.1007/s11263-007-0075-7. Disponível em: <<https://doi.org/10.1007/s11263-007-0075-7>>. Citado 1 vez na página 12.

TOBII. **What is eye tracking**. [S.l.: s.n.], 2019. <https://www.tobii.com/>. Acessado: 10 maio 2019. Citado 1 vez na página 12.

VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. In: PROCEEDINGS of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001. [S.l.: s.n.], dez. 2001. v. 1, p. i–i. DOI: 10.1109/CVPR.2001.990517. Citado 6 vezes nas páginas 18, 21–23.

ZHANG, X. et al. Graph Based Discriminative Learning for Robust and Efficient Object Tracking. In: 2007 IEEE 11th International Conference on Computer Vision. [S.l.: s.n.], out. 2007. p. 1–8. DOI: 10.1109/ICCV.2007.4409034. Citado 1 vez na página 12.