

# Método Computacional para Medição Automática do Diâmetro Limbar

## Computational Method for Automatic Measurement of the Limbus Diameter

João Vitor Miranda Roma  
Universidade Federal do  
Maranhão - UFMA  
Av. dos Portugueses, 1966  
Bacanga - CEP 65080-805  
+55 98 98160 4895  
jvmr2.0@gmail.com

João Dallyson Sousa de  
Almeida  
Universidade Federal do  
Maranhão - UFMA  
Av. dos Portugueses, 1966  
Bacanga - CEP 65080-805  
+55 98 3272 9280  
jdallyson@ufma.br

Jorge Antonio Meireles  
Teixeira  
Universidade Federal do  
Maranhão - UFMA  
Av. dos Portugueses, 1966  
Bacanga - CEP 65080-805  
+55 98 3301 9611  
jorgemeireles1@bol.com.br

Geraldo Braz Junior  
Universidade Federal do  
Maranhão - UFMA  
Av. dos Portugueses, 1966  
Bacanga - CEP 65080-805  
+55 98 3272 9246  
geraldo.braz@ufma.br

Anselmo Cardoso de  
Paiva  
Universidade Federal do  
Maranhão - UFMA  
Av. dos Portugueses, 1966  
Bacanga - CEP 65080-805  
+55 98 3301 9246  
paiva@nca.ufma.br

### ABSTRACT

The measurement of the limbus diameter in millimeters is useful for ophthalmologists in various tests, such as those that enable the detection of congenital glaucoma. Some exams require the patient to interact with the doctor, providing information during the exam. Patients who cannot collaborate, such as children aged 0-3, need to be sedated to allow the specialist to check the diameter of the limbus. This measurement is not always accurate, because in medical practice it is common to use a ruler close to the eye to gauge the diameter of the limbus. In this context, it is appropriate to develop a computational solution that avoids the use of invasive techniques in patients, also avoiding the need to sedate them for such examinations, as well as improving the accuracy of the measurement. In this work, a computational method is proposed for the automatic detection of limbus in patient images and for the calculation of its diameter in millimeters. The results obtained by the developed method are compatible with the values obtained by the manual measurement method. The performance obtained by the developed technique indicates that the proposed methodology has potential for application in ophthalmic offices.

### Keywords

Limbus Diameter, Hough Transform, Raspberry Pi

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

### RESUMO

A medida em milímetros do diâmetro do limbo é útil para os médicos oftalmologistas em diversos exames, como os que possibilitam a detecção do glaucoma congênito. Alguns exames precisam que o paciente interaja com o médico, fornecendo informações durante a realização do exame. Os pacientes que não podem colaborar, como crianças de 0 a 3 anos, precisam ser sedados para permitir que o especialista verifique o diâmetro do limbo. Esta medida nem sempre é precisa, pois na prática médica, é comum a utilização de uma régua próxima ao olho para aferir o diâmetro do limbo. Neste contexto, faz-se oportuno o desenvolvimento de uma solução computacional que evite a utilização de técnicas invasivas nos pacientes, evitando também a necessidade de sedá-los para a realização de tais exames, assim como melhorando a precisão da medida. Neste trabalho, é proposto um método computacional para a detecção automática do limbo em imagens de pacientes e para o cálculo do seu diâmetro em milímetros. Os resultados obtidos através do método desenvolvido são compatíveis com os valores obtidos pelo método manual de medição. O desempenho obtido pela técnica desenvolvida indica que a metodologia proposta tem potencial de aplicação em consultórios oftalmológicos.

### Palavras-chave

Diâmetro Limbar, Transformada de Hough, Raspberry Pi

### CCS Concepts

•Applied computing → Health care information systems;

## 1. INTRODUÇÃO

O olho humano é o órgão responsável pela visão humana. Caracterizado pelo formato globular, esse órgão está localizado em uma cavidade óssea na face e suas estruturas podem ser divididas em três camadas: externa, composta pela esclera, córnea e limbo; medial, onde está a íris, pupila, coróide e corpo ciliar; e interna, composta pela retina. A Figura 1 apresenta os principais componentes da camada externa do olho, cujo conhecimento prévio é importante para a compreensão deste trabalho [20].

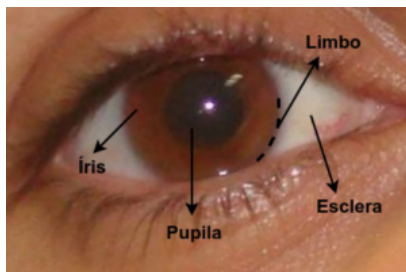


Figure 1: Estrutura externa do olho humano.

A medida em milímetros do diâmetro do limbo é útil para os médicos oftalmologistas em certos exames, como os que possibilitam a detecção do glaucoma congênito. Crianças com glaucoma congênito unilateral possuem um olho maior que o outro, e conseqüentemente, possuem um limbo maior que o outro, apresentando diâmetro corneano horizontal maior que 12 milímetros [6]. Crianças com glaucoma congênito bilateral possuem os dois olhos com o tamanho alterado, fora do normal. Assim, é possível detectar a presença ou ausência do glaucoma nesse grupo de pacientes por meio da mensuração do diâmetro limbar.

O Glaucoma é uma designação genérica a um grupo de doenças oculares distintas que provocam danos ao nervo óptico e perda da visão [11]. O glaucoma congênito é a causa de 16% dos casos de crianças cegas e 15% dos casos de crianças com baixa visão no Brasil [3]. Sendo assim, faz-se necessário o diagnóstico precoce de tal patologia.

No entanto, os exames para detectar o glaucoma geralmente necessitam da colaboração do paciente. Alguns exames exigem um posicionamento correto e estático do corpo e dos olhos durante sua realização, outros exames, precisam que o paciente interaja com o médico, fornecendo informações durante a realização do exame [7]. Nesse contexto, existem dificuldades em aplicar os métodos usuais em crianças de 0 a 3 anos que não conseguem colaborar da maneira necessária durante os exames [21].

Na prática médica, os pacientes que não podem colaborar precisam ser sedados para permitir que o especialista verifique o tamanho do limbo. Esta medida nem sempre é precisa, pois o mesmo utiliza uma régua próxima ao olho para aferir o diâmetro limbar, como exibido na Figura 2.

A prática médica atual para a medida do diâmetro do limbo em pacientes não colaborativos, consiste em sedar o paciente e utilizar de técnicas invasivas e nem sempre precisas para o aferir este valor. Neste contexto, faz-se oportuno o desenvolvimento de uma solução computacional que evite a utilização de técnicas invasivas nos pacientes, evitando também a necessidade de sedá-los para a realização de tais exames, assim como melhorando a precisão na aferição



Figure 2: Medição do limbo em paciente sedado.

da medida, uma vez que métodos computacionais tendem a ser mais precisos do que métodos manuais, pois estes são sensíveis a possíveis falhas humanas do especialista, devido a alta necessidade de coordenação motora e precisão visual na realização do exame.

A detecção do limbo em imagens possui diversas aplicações ligadas à segmentação da íris como região de interesse. Algumas dessas aplicações são: exames como o teste do reflexo vermelho e testes de detecção do estrabismo, rastreamento da posição da íris em vídeos e reconhecimento biométrico com base no padrão visual da íris.

Diante do contexto apresentado, o objetivo deste artigo é apresentar um método computacional para aferir automaticamente o diâmetro do limbo em milímetros, com base em fotos de pacientes, utilizando técnicas de processamento de imagens digitais, visando ser utilizado para que os médicos oftalmologistas consigam obter esta medida de forma menos invasiva e também para que possa facilitar a realização de exames onde esta medida seja necessária.

A principal contribuição deste trabalho está na proposta de um método automático associada a um hardware de baixo custo aferir automaticamente o diâmetro limbar.

### 1.1 Trabalhos Relacionados

Esta seção apresenta pesquisas, equipamentos e trabalhos relacionados à localização dos olhos em imagens, detecção automática da íris ou do limbo e processamento de imagens no *Raspberry Pi*. Tais estudos foram encontrados utilizando conceitos parciais da técnica de revisão sistemática da literatura em conjunto com o portal Google Acadêmico<sup>1</sup>.

Lorenzi (2014) implementou um *software* de detecção de estrabismo para dispositivos móveis, utilizando técnicas de processamento de imagem como o classificador em cascata de Haar [14] e a transformada de Hough [10], aplicada a detecção de círculos. O classificador em cascata de Haar foi usado para detectar a região da face e dos olhos e a transformada teve como objetivo detectar a região do limbo e do brilho dos olhos.

Cruz et al. (2016) desenvolveram um dispositivo que utiliza um *Raspberry Pi* e que executa o reconhecimento de íris por meio de processamento de imagem. Realiza funções de pré-processamento antes de enviar as imagens à aplicação para contornar as limitações do *Raspberry Pi*. Já Kunik et al. (2017) apresentaram um sistema completo para reconhecimento de íris. O sistema proposto é baseado no computador de placa única *Raspberry Pi 3B*, com ênfase na funcionalidade do processo de reconhecimento como um todo,

<sup>1</sup><https://scholar.google.com.br/>

incluindo aspectos de hardware e software. Uma aquisição de dados por infravermelho permitiu uma segmentação e codificação eficiente da íris. Ambos os trabalhos utilizam a Transformada de Hough para detecção e localização da íris.

Almeida et al. (2015) propuseram um método para detectar estrabismo, utilizando SVM (do inglês: *support vector machine*, ou máquina de vetores de suporte) para detectar os olhos e a transformada de Hough para detectar a região circular do limbo. Silveira et al. (2018) propuseram um dispositivo portátil e de baixo custo baseado em uma placa *Raspberry Pi*, com funcionalidades semelhantes aos dispositivos *Photoscreeners*<sup>2</sup> disponíveis no mercado. Tem o propósito de incorporar diferentes métodos computacionais para detectar patologias da visão.

Utilizando técnicas semelhantes aos trabalhos citados, este trabalho apresenta um método para medir automaticamente o limbo, problema ainda não abordado na literatura e sem ferramentas simples e de baixo custo para tal aferição.

O trabalho está organizado em 5 seções. A seção 2 contém a fundamentação teórica sobre os conceitos que foram utilizados durante o desenvolvimento do trabalho. Na seção 3 é apresentada a metodologia proposta por este trabalho para a detecção do limbo em imagens de pacientes e para o cálculo da sua mensuração em milímetros. Na seção 4 são mostrados os resultados alcançados por meio da aplicação do método e hardware propostos. Por fim, na seção 5 são expressadas as conclusões e sugestões de trabalhos futuros.

## 2. FUNDAMENTAÇÃO TEÓRICA

### 2.1 Imagem Digital

Uma imagem digital é uma função bidimensional de intensidade de luz  $f(x, y)$ , onde  $x$  e  $y$  são as coordenadas espaciais e o valor de  $f$  em qualquer ponto  $(x, y)$  é proporcional ao brilho da imagem naquele ponto. Também pode ser considerada como sendo uma matriz cujos índices de linha e colunas identificam um ponto na imagem. Este ponto é chamado de pixel [8]. Em ciência da computação, o processamento digital de imagens é o uso de algoritmos de computador para realizar o processamento em imagens digitais [2].

#### 2.1.1 Histograma

A representação gráfica da distribuição dos valores de tonalidade dos pixels em uma imagem digital é chamada de histograma da imagem. Ele mostra o número de pixels para cada valor de tom, ou seja, o eixo horizontal do gráfico representa a variação de tons, enquanto o eixo vertical representa o número total de pixels nesse tom específico, em outras palavras, o lado esquerdo do eixo horizontal representa as regiões mais escuras da imagem e o lado direito representa as regiões mais claras, o eixo vertical representa o número total de pixels (tamanho da área) localizada em cada uma dessas regiões.

### 2.2 Características de Haar

Viola e Jones (2001) propuseram uma abordagem de aprendizagem de máquina para detecção de objetos visuais em imagens. Nos estágios iniciais do algoritmo, são utilizadas características chamadas de *Haar-like features*, descritas previamente em [18], que possuem esse nome pois são baseadas na Transformada de Haar [9].

<sup>2</sup>Dispositivos que através da análise de imagens são capazes de detectar patologias da visão.

Uma *Haar-like feature* simples pode ser definida como a diferença entre os somatórios dos valores dos pixels nas áreas dentro de retângulos na imagem. Estes retângulos podem ter qualquer tamanho e estarem em qualquer posição dentro da imagem. Podem ser utilizados conjuntos de dois, três ou quatro retângulos. Os valores encontrados indicam certas características de uma área específica da imagem. Cada tipo de característica pode indicar a existência (ou ausência) de algo na imagem, como bordas ou mudanças na textura. Um conjunto de dois retângulos pode detectar uma borda entre uma região escura e uma região clara na imagem, por exemplo. O detector de objetos foi aperfeiçoado por Lienhart e Maydt (2002), adicionando a detecção de *Haar-like features* rotacionadas em 45°, melhorando a detecção das características.

Para a tarefa de detecção de faces, a primeira característica selecionada se concentra na região dos olhos, que muitas das vezes é mais escura do que a região do nariz e bochechas (Figura 3). A segunda característica selecionada se baseia no fato dos olhos serem mais escuros do que o nariz. A primeira característica mede a diferença de intensidade entre a região dos olhos e a região superior das bochechas. A segunda característica compara as intensidades nas regiões dos olhos com a intensidade no nariz [24].

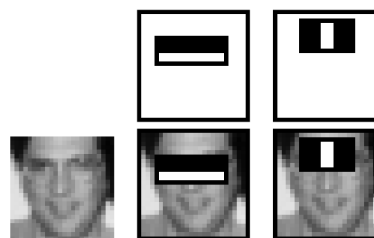


Figure 3: Comparações de intensidade de brilho entre regiões.

Neste trabalho, utilizou-se o detector de características de Haar para detectar as regiões dos olhos nas imagens dos pacientes.

### 2.3 Suavização de Imagens

Vários filtros utilizados em processamento de sinais, como filtros passa-alta, passa-baixa, etc., podem ser aplicados às imagens. Filtros passa-baixa ajudam a remover ruídos da imagem, desfocá-la, enquanto filtros passa-alta ajudam a encontrar bordas nas imagens. O desfoque (suavização) da imagem é obtido pela convolução da imagem com um *kernel* de filtro passa-baixa [16].

A saída de um filtro espacial linear de suavização é simplesmente a média dos pixels contidos na vizinhança da máscara do filtro. Os filtros de suavização substituem o valor de cada pixel em uma imagem pela média dos níveis de cinza na vizinhança definida pela máscara de filtro. Esse processo resulta em uma imagem com as transições suavizadas. Como o ruído aleatório consiste tipicamente em transições bruscas nos níveis de cinza, a aplicação mais óbvia de suavização é a redução de ruído [8].

O procedimento consiste em percorrer uma matriz de convolução sobre a imagem. Acima de cada pixel, no exemplo de uma matriz  $5 \times 5$ , são adicionados todos os 25 pixels abaixo desta matriz, é calculado sua média e é substituído o valor do

pixel central pelo novo valor médio. A operação é repetida para todos os pixels na Imagem [16].

$$K = \frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Neste trabalho, o filtro da média é utilizado visando a redução do ruído como parte do tratamento realizado nas imagens processadas.

## 2.4 Limiarização de Otsu

Limiarização (*threshold*) é a técnica mais simples de segmentação de imagem. A partir de uma imagem em escala de cinza, o limiar pode ser usado para criar imagens binárias [19]. O método de limiarização mais simples substitui cada pixel em uma imagem por um pixel preto, se a intensidade da imagem na posição  $I_{i,j}$  for menor que alguma constante fixa  $T$  (*threshold*), ou seja, se  $I_{i,j} < T$ , ou por um pixel branco, se a intensidade da imagem for maior que essa constante.

O método de Otsu [17] é usado para encontrar automaticamente um valor para o limiar (*threshold*). O algoritmo assume que a imagem contém duas classes de pixels, seguindo o histograma bimodal (pixels de primeiro plano e pixels de fundo), então tenta encontrar um valor que minimize a variância ponderada entre as classes. Ou seja, ele encontra um valor de  $t$  que fica entre os dois picos das classes no histograma, de modo que as variações para ambas as classes são mínimas. Neste trabalho, o Limiar de Otsu é aplicado para segmentar de forma otimizada as regiões da íris e da esclera nas imagens de olhos.

## 2.5 Detecção de Bordas de Canny

Os pontos nos quais o brilho de uma imagem muda drasticamente, acentuadamente ou tem descontinuidades, são tipicamente organizados em um conjunto de segmentos de linhas curvas chamados de bordas. A detecção de bordas inclui uma variedade de métodos matemáticos que visam identificar tais mudanças [23].

Dentre as técnicas de detecção de bordas, o detector de Bordas de Canny [1] é um algoritmo de detecção de borda baseado no Filtro de Sobel [22]. Basicamente utiliza dois limites diferentes no Filtro de Sobel, um baixo e um alto, o limite mais baixo permite passar mais bordas, portanto, mais bordas do que o necessário são detectadas. Ao passar a imagem por um segundo limite maior, apenas as arestas que pertencem aos contornos mais importantes são detectadas. O algoritmo de Canny combina esses dois mapas de borda para produzir contornos “ótimos”. Ele opera mantendo apenas os pontos das bordas do limite baixo, para o qual existe um caminho contínuo de bordas, vinculando esses pontos a uma borda pertencente ao limite alto [13]. Neste trabalho, o detector de bordas de Canny é utilizado para encontrar os contornos que compõem os olhos nas imagens processadas, em busca de realçar a região do limbo.

## 2.6 Detector de círculos de Hough

A detecção de objetos circulares é uma tarefa de extração de características fundamentais no reconhecimento de padrões, que foi continuamente desenvolvida para obter melhor desempenho e precisão computacional. Diversas aplicações dentro dos campos de: visão computacional, física e biologia,

dependem da detecção de padrões circulares, por exemplo, detecção de íris, contagem de células, identificação de formas de células e rastreamento de partículas [25].

A transformada de Hough (TH) [10] foi elaborada para detectar características analiticamente representáveis em imagens, como linhas, círculos e elipses. Para realizar uma análise baseada em conteúdo de uma imagem, é necessário extrair recursos significativos da coleção de pixels que constituem a imagem. Contornos, linhas, regiões e assim por diante são elementos de imagem fundamentais que definem o conteúdo de uma imagem [13]. A transformada é aplicada após detectar as bordas da imagem, por exemplo, pelo detector de bordas de Canny [15]. A equação paramétrica, usada pela transformada de Hough para detecção de círculos é apresentada a seguir.

$$\begin{aligned} x &= a + r \cdot \cos(\theta) \\ y &= b + r \cdot \sin(\theta) \end{aligned}$$

A equação paramétrica do círculo possui três parâmetros,  $a$  e  $b$  representando o centro do círculo e  $r$  representando o raio. Quando o ângulo  $\theta$  passar por todos os 360 graus, os pontos  $(x, y)$  traçam o perímetro de um círculo. O trabalho da transformada de Hough é desenhar o perímetro de um círculo para cada ponto de borda encontrado, no espaço de Hough, como pode ser visto na Figura 4.

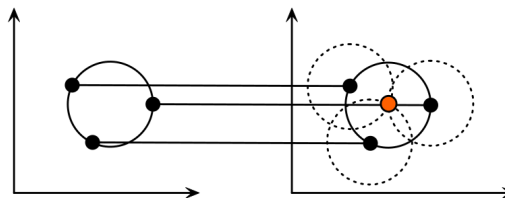


Figure 4: Esquemática do funcionamento da transformada de Hough.

Na Figura 4, cada ponto de borda no espaço geométrico (lado esquerdo) gera o traço de um círculo no espaço de Hough (lado direito). Os traços de círculos gerados no espaço de Hough tem um ponto de intersecção que representam a coordenada  $(a, b)$ , que por sua vez, representa o centro de um círculo na imagem original, com raio igual aos círculos traçados no espaço de Hough [15]. Neste trabalho, foi utilizada uma versão modificada da TH descrita na Metodologia (Seção 3.1.1).

## 2.7 Métricas de Avaliação

Na análise estatística de conjuntos binários, algumas métricas são utilizadas para avaliar um classificador. Em particular, tratando-se de segmentação de imagens, é possível utilizar tais métricas para medir o desempenho de um algoritmo de segmentação, utilizando-as para comparar a segmentação encontrada pelo algoritmo e uma imagem de referência que indique a segmentação correta que está se tentando alcançar.

Classificando as duas classes dos conjuntos binários como positivos e negativos, no contexto da segmentação de imagens, um pixel que está na região que se quer segmentar pode ser classificado como parte dos positivos, e um pixel que está fora desta região é classificado como parte dos negativos.

Na imagem resultante do método de segmentação que está

sendo avaliado, um pixel que foi classificado como negativo, é classificado como um verdadeiro negativo (VN) caso, na imagem de referência, o pixel na mesma posição é classificado como negativo, e é classificado como um falso negativo (FN) caso, na imagem de referência, conste que o pixel naquela posição deveria ser classificado como positivo.

Analogamente, se um pixel da imagem resultante foi classificado como positivo pelo algoritmo de segmentação, será classificado como um verdadeiro positivo (VP) caso, na imagem de referência, o pixel nesta mesma posição foi classificado como positivo, e será classificado como um falso positivo (FP) caso, na imagem de referência, o pixel naquela posição seja classificado como negativo.

As métricas comumente utilizadas para validar resultados de classificação são: a Sensibilidade (S), a Especificidade (E), a Precisão (P) e a Acurácia (A). A Sensibilidade corresponde à proporção de itens do conjunto classificados corretamente como positivos (verdadeiros positivos) em relação aos itens de fato positivos (verdadeiros positivos e falsos positivos).

$$S = \frac{VP}{VP + FN}$$

A Especificidade trata-se do análogo da sensibilidade, corresponde à proporção de itens classificados corretamente como negativos (verdadeiros negativos) para os todos os itens de fato negativos (verdadeiros negativos e falsos positivos).

$$E = \frac{VN}{VN + FP}$$

A Precisão fornece a proporção de itens corretamente classificados como positivos (verdadeiros positivos) em relação a todos os itens classificados como positivos (verdadeiros positivos e falsos positivos).

$$P = \frac{VP}{VP + FP}$$

A Acurácia corresponde à taxa total de acerto do classificador, ou seja, qual a porcentagem de itens corretamente classificados (verdadeiros positivos e verdadeiros negativos) em relação ao total de itens.

$$A = \frac{VP + VN}{VP + FP + VN + FN}$$

O Coeficiente de Dice é uma estatística utilizada para medir a similaridade entre dois conjuntos. Pode ser descrito por:

$$DSC = \frac{2VP}{2VP + FP + FN}$$

As métricas descritas são empregadas na avaliação do método de detecção do limbo proposto neste trabalho, utilizando como referência, uma base de imagens faciais onde a localização e tamanho dos limbos foram marcadas manualmente por um especialista. Os resultados foram comparados com as métricas de desempenho de outros detectores de círculos aplicados à mesma base de imagens.

### 3. MÉTODO E HARDWARE PROPOSTO

O método proposto para a medição automática do diâmetro do limbo em imagens de pacientes consiste nas etapas apresentadas na Figura 5 e descritas a seguir.

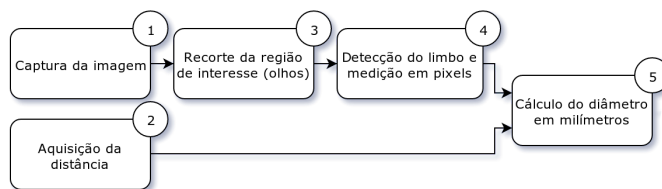


Figure 5: Etapas do método proposto.

Na 1ª etapa é realizada a captura da imagem do paciente e, em paralelo, é executada a 2ª etapa, onde a distância entre a câmera e o rosto do paciente no momento da fotografia é aferida, isto é feito utilizando um sensor de distância acoplado à câmera, acionando-o no momento da fotografia, o valor registrado deve ser armazenado para uso posterior. Tal medida também pode ser feita utilizando uma estrutura onde a câmera e o rosto do paciente fiquem em posições estáticas e a distância entre eles possa ser medida. Esta medida é necessária para calcular o diâmetro do limbo em milímetros, a sua utilização é explicada em detalhes na subseção Medição do Limbo (Subseção 3.2).

#### 3.1 Detecção do limbo

Anteriormente à detecção do limbo realiza-se o recorte da região dos olhos na foto facial. Para esta fase (3ª etapa), foi utilizado o algoritmo de [24] (Subseção 2.2) para a detecção de características semelhantes a olhos na imagem. Procuram-se as regiões de cada olho na imagem levando em consideração algumas restrições de tamanho em relação às dimensões da imagem para evitar falsos positivos, por exemplo, cada uma das duas regiões detectadas deve ocupar no máximo metade e no mínimo um décimo do comprimento da imagem, como a imagem de entrada do método é facial, caso a região retornada pelo algoritmo como possível olho seja maior ou menor do que o intervalo supracitado, certamente trata-se de uma má detecção. Após encontradas, as regiões de interesse, são recortadas da imagem inicial em duas novas imagens que serão processadas no restante do método.

Para a detecção do limbo, a 4ª etapa, cada uma das duas imagens dos olhos é recortada novamente para que o foco seja apenas na esclera e na íris. Aqui, busca-se recortar a imagem o menor possível para que apenas a esclera e íris sejam exibidas, com a presença do mínimo de pele na imagem, pois o tom de cinza referente à pele na imagem interfere no Limiar de Otsu, que utiliza o tom de cinza mais claro presente na esclera e o mais escuro presente na íris no cálculo do valor otimizado de *threshold*. Mas ao mesmo tempo, o recorte deve ser grande o suficiente para que a íris esteja completamente inserida nele para que, na imagem resultante, as bordas do limbo sejam completamente detectadas. A forma de corte utilizada foi de dividir a imagem em três partes iguais na horizontal e em quatro partes iguais na vertical, e então a região buscada é a compreendida entre o segundo terço da divisão horizontal e os segundo e terceiro quartos da divisão vertical. A técnica é ilustrada pela Figura 6.

A extração das bordas da imagem é objetivada para que



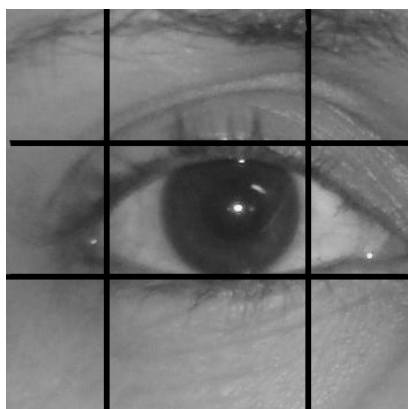


Figure 6: Recorte da imagem dos olhos.

seja executado o detector de círculos sobre a imagem resultante. No entanto, se apenas for aplicado o detector de bordas, a imagem resultante possuirá bordas provenientes de ruídos na imagem que reduzem a eficiência do detector de círculos. Portanto, para melhores resultados, é realizado um tratamento na imagem antes da extração das bordas. Os passos são ilustrados pela Figura 7. Após o recorte, cada imagem passa primeiramente por um processo de redução de ruído, por meio da suavização (Subseção 2.3) utilizando o filtro da média do OpenCV<sup>3</sup> com os parâmetros informando o uso de uma máscara de 5x5. Como normalmente a região da íris é mais escura e a região da esclera é mais clara, e o limbo que pretende-se detectar é a fronteira entre essas duas regiões, o seu contorno será aguçado ao segmentar a imagem com base nessa mudança de tons, por isso, em seguida é aplicado o limiar de Otsu (Subseção 2.4), que com base nos dois tons predominantes no histograma da imagem, encontra um valor de *threshold* otimizado para segmentar as regiões (Figura 8).

Uma vez encontrada a imagem binarizada de cada olho, por fim é executado o detector de bordas de Canny (Subseção 2.5) na imagem, encontrando as fronteiras entre as regiões preta e o branca da imagem. Foram utilizados parâmetros proporcionais ao valor de *threshold t* encontrado pelo Limiar de Otsu, buscando uma extração bastante justa dos contornos, os valores de limite inferior e superior do detector de Canny foram respectivamente  $0,99t$  e  $1,01t$ . Como o contorno do limbo havia sido realçado pelo limiar, ao serem extraídas as bordas da imagem binarizada, o contorno do limbo estará entre as bordas encontradas.

### 3.1.1 Modificações no Detector de Círculos de Hough

No caso específico da detecção de círculos em uma imagem composta de bordas provenientes dos contornos de uma imagem de olho, onde se busca encontrar o círculo correspondente ao limbo, podem ocorrer detecções incorretas devido ao fato de que outras regiões do olho também possuem formas circulares e estarem presentes na imagem das bordas, como as pálpebras.

Desta forma, é conveniente implementar uma modificação no algoritmo de detecção de círculos de Hough, de modo que, ao percorrer os pontos da borda no espaço geométrico, gere no espaço de Hough apenas os pontos no contorno de um

<sup>3</sup><https://opencv.org/>

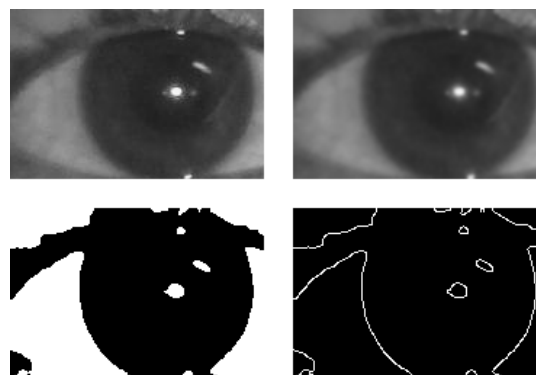


Figure 7: Etapas do tratamento da imagem para a detecção do limbo.

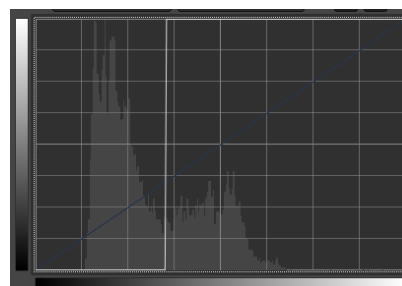


Figure 8: Aplicação do Limiar de Otsu no histograma.

círculo dentro de um determinado intervalo de ângulos desejado, como exemplificado na Figura 9, onde apenas pontos que correspondem a uma abertura lateral de  $120^\circ$  no círculo são levados em consideração ao se gerar o acumulador.

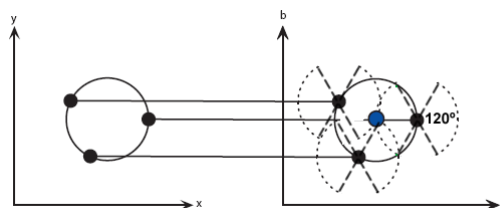
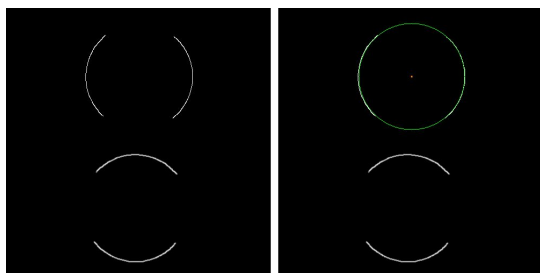


Figure 9: Modificação no Detector de Círculos de Hough.

Deste modo, quando um ponto de acumulação no espaço de Hough é encontrado, ele representa a coordenada de um centro de círculo na imagem original, porém, apenas os pontos do círculo dentro do intervalo de ângulos desejado foram levados em consideração no acumulador. Em outras palavras, se uma forma circular na imagem tiver, em sua maioria, pontos dentro do intervalo desejado, ela será detectada pelo algoritmo e se ela tiver, em sua maioria, pontos fora do intervalo desejado, ela não será detectada.

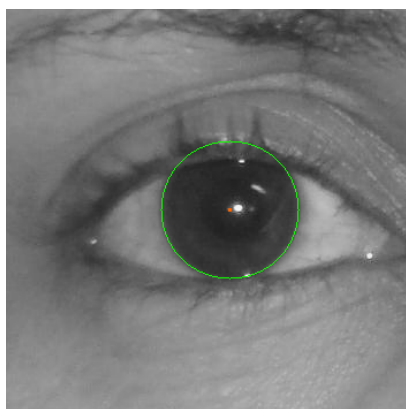
Na Figura 10 o funcionamento da modificação é demonstrado de forma simples. Duas formas circulares de teste foram desenhadas, uma com pontos em sua maioria dentro do intervalo de ângulos desejado, compreendendo as laterais do círculo, e outra com pontos em sua maioria fora do in-

tervalo desejado, e como esperado, apenas a primeira forma foi detectada como círculo.



**Figure 10: Aplicação da modificação no Detector de Círculos de Hough.**

Na imagem das bordas, é aplicado o detector de círculos de Hough modificado, utilizando o intervalo de raios de um quarto a um oitavo da largura da imagem da região do olho, e os pontos nos intervalos de  $135^\circ$  a  $225^\circ$ , e de  $315^\circ$  a  $45^\circ$ , correspondendo a uma abertura de  $90^\circ$  de cada lado do círculo, valores advindos da simples divisão do círculo em quatro setores iguais e selecionando-se os dois setores laterais. Ao encontrar na imagem das bordas o círculo que sobrepõe o limbo, armazena-se o seu diâmetro em pixels. A Figura 11 demonstra uma detecção do limbo realizada seguindo os passos descritos.



**Figure 11: Limbo detectado.**

### 3.2 Medição do limbo

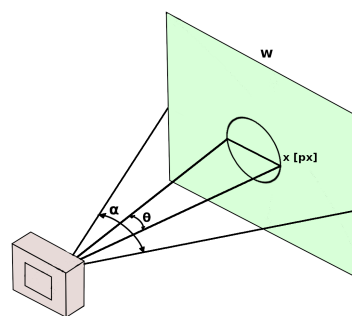
O tamanho de um objeto em pixels em uma imagem não tem relação direta com seu tamanho real, uma vez que, de acordo com variações de características entre diferentes câmeras, como resolução, distância focal, etc., ou até mesmo com a variação da distância entre a câmera e o objeto fotografado, objetos de um determinado tamanho podem ter diferentes medidas em pixels em diferentes imagens. Para que a medida do limbo seja armazenada e reportada na unidade de milímetros, como é objetivo deste trabalho, faz-se necessária uma conversão do diâmetro entre pixels e milímetros (5ª etapa).

O ângulo do campo de visão de uma câmera pode ser encontrado diretamente em sua documentação, nos metadados das imagens tiradas por ela ou pode ser calculado com base em informações presentes nestes utilizando a Equação

1, onde  $\alpha$  representa o ângulo do campo de visão,  $f$  representa a distância focal, que é a distância entre a lente e o sensor digital e  $s$  representa a dimensão do sensor, na direção horizontal, vertical ou diagonal, dependendo da direção do ângulo que se busca determinar.

$$\tan(\alpha/2) = \frac{s/2}{f} \Rightarrow \alpha = 2 \cdot \arctan\left(\frac{s/2}{f}\right) \quad (1)$$

Na Figura 12, o ângulo  $\theta$  representa a fração do ângulo do campo de visão da câmera ( $\alpha$ ) que compreende o tamanho do que está sendo medido. É possível descobrir o ângulo  $\theta$  por meio de uma regra de três simples, descrita na Equação 2, utilizando o diâmetro do limbo em pixels ( $x$ ), a resolução da câmera utilizada em pixels ( $w$ ) na direção horizontal, vertical ou diagonal, dependendo da direção em que se está sendo feita a medição e o ângulo do campo de visão da câmera ( $\alpha$ ), em graus, encontrado pela Equação 1.



**Figure 12: Fração do ângulo de visão da câmera que compreende o objeto medido.**

$$\frac{\theta}{\alpha} = \frac{x}{w} \Rightarrow \theta = \alpha \cdot \frac{x}{w} \quad (2)$$

A técnica de conversão proposta necessita que no momento da captura da imagem, também se capture a distância entre a câmera e o rosto do paciente utilizando um sensor de distância acoplado a câmera, por exemplo. Com base nesta medida e no tamanho do limbo em pixels na imagem, o tamanho em milímetros pode ser calculado por meio de técnicas de trigonometria descritas na Equação 3 e demonstradas geometricamente na Figura 13.

Na Figura 13, o ponto  $C$  representa a câmera,  $\theta$  representa o ângulo de visão da câmera que o objeto que está sendo medido ocupa, encontrado pela Equação 2, os pontos  $A$  e  $B$  representam as extremidades do que está sendo medido, sendo  $x$  o tamanho do que está sendo medido em milímetros, a medida que se deseja encontrar, e  $d$  representa a distância entre a câmera e o que está sendo medido no momento da fotografia.

$$\tan(\theta/2) = \frac{x/2}{d} \Rightarrow x = 2 \cdot d \cdot \tan(\theta/2) \quad (3)$$

### 3.3 Dispositivo para detecção de patologias da visão

O equipamento utilizado neste trabalho foi o dispositivo portátil de baixo custo, baseado em uma placa *Raspberry*

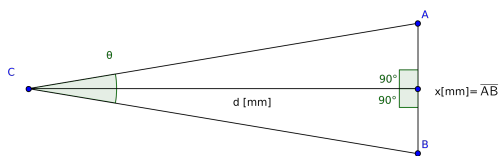


Figure 13: Representação geométrica.

*Pi 3 B* proposto por [21]. O componente responsável pela aquisição das imagens é a Câmera *Raspberry Pi v2*, de 8 MP (*megapixels*), com foco fixo e resolução de 3280 x 2464 pixels. Caso necessário, pode ser utilizada a versão NoIR, esta versão da câmera não possui filtro infravermelho, o que permite capturar luz infravermelha nas fotografias, necessária em alguns exames oftalmológicos que utilizam iluminação com esta frequência.

O dispositivo possui uma placa de circuito que dispõe de LEDs (*Light Emitting Diode*) de alto brilho e LEDs infravermelhos posicionados formando um círculo de aproximadamente 4,5 cm de diâmetro ao redor da câmera. A placa foi desenvolvida para exercer a função de fornecer iluminação para a aquisição de imagens através da câmera em ambientes de baixa luminosidade, assim como possibilitar exames semelhantes aos realizados pelos dispositivos disponíveis no mercado, que utilizam o reflexo do infravermelho nos olhos dos pacientes para identificar patologias da visão.

O envólucro que organiza os componentes eletrônicos possui suportes para o encaixe das mãos do operador com um formato ergonômico. Na lateral, encontram-se aberturas para as portas do *Raspberry*. O tamanho e formato do dispositivo busca reproduzir o padrão dos equipamentos disponíveis no mercado. Para a interação homem-máquina, o componente utilizado é um *display* de 5 polegadas *touchscreen*. Há uma interface gráfica desenvolvida em Python, que é responsável pela realização dos exames e todo o processo de cadastro de pacientes e métodos computacionais.

O dispositivo foi concebido com o propósito de permitir a inclusão de diversos métodos computacionais para detectar patologias da visão. O método que está sendo apresentado neste estudo foi incorporado no dispositivo. Para os fins deste trabalho, foi necessário realizar algumas modificações no dispositivo. Para medir a distância entre o dispositivo e o rosto do paciente, foi escolhido utilizar o Sensor Ultrassônico HC-SR04, devido ao seu baixo custo e facilidade de integração com o *Raspberry Pi*. O sensor emite um som numa frequência acima da faixa audível por seres humanos, e capta o eco. Com base no tempo decorrido entre o envio e o recebimento do som, e sabendo-se a velocidade do som no ar, é possível calcular a distância percorrida.

Para integrar o sensor de distância ao dispositivo, foi desenvolvida uma nova placa de circuito. Nesta placa, os LEDs de luz branca e os de luz infravermelha foram dispostos de uma forma semelhante à da placa original do dispositivo, em forma de círculo ao redor da câmera. A versão modificada do dispositivo pode ser vista na Figura 14.

Na Tabela 1 são apresentadas as especificações da câmera do dispositivo, importantes para este trabalho. Na Figura 15 é esquematizada a relação entre o campo de visão total da lente da câmera (círculo externo) e a região do campo de fato é registrada na fotografia, compreendida no retângulo



Figure 14: Dispositivo com nova placa de circuito.

interno, pois corresponde à parte da luz captada que de fato interfere com o sensor da câmera.

Table 1: Especificações de Hardware da Câmera *Raspberry Pi v2*

PARÂMETRO	VALOR
Resolução da câmera	3280 2464 pixels
Dimensões do sensor	3,68 x 2,76 mm (4,6 mm diagonal)
Distância focal	3,04 mm

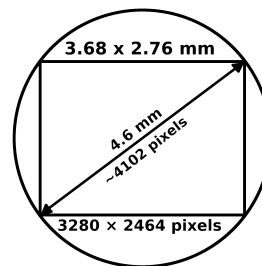


Figure 15: Relação entre o campo de visão da câmera e a região da fotografia.

A medida diagonal do sensor é utilizada pois é a que mais se aproxima da totalidade do campo de visão da lente. Ao substituir tais valores nas Equações 1, 2 e 3, obtemos a Equação 4, que será utilizada neste trabalho, onde  $x[px]$  é o tamanho do que se está medindo em pixels na imagem,  $d$  é a distância em milímetros entre a câmera e o que se está medindo e  $x[mm]$  é a medida que se deseja encontrar, o tamanho do que está sendo medido na unidade milímetros.

$$x [mm] = 2 \cdot d \cdot \tan \left( \frac{x [px]}{2} \cdot 0,01809203817 \right) \quad (4)$$

## 4. RESULTADOS E DISCUSSÃO

Esta seção apresenta os resultados obtidos pela fase de detecção do limbo executada em uma base de imagens de rostos e são discutidas as métricas de desempenho calculadas com



base nas imagens resultantes. Também são apresentados e discutidos os resultados da aplicação do método proposto para o cálculo do diâmetro do limbo em milímetros, em um experimento em ambiente controlado utilizando réplicas de olhos e em fotos de pacientes reais.

Os experimentos foram realizados em imagens de pacientes adquiridas em um consultório oftalmológico especializado em estrabismo localizado na cidade de São Luís-MA. Todos os pacientes concordaram em participar do estudo. Para validar o método, foram utilizadas também bases de imagens públicas que possuíam as regiões de interesse deste estudo.

#### 4.1 Detecção do limbo em uma base de imagens

Os parâmetros propostos no método de detecção do limbo foram testados em uma base de imagens de face, utilizada em [5], que possui imagens de 45 pacientes de estrabismo em diversos posicionamentos, totalizando 248 fotos. Destas, foram utilizadas apenas imagens dos pacientes sem óculos e em posições frontais em relação à câmera, pois estas são as que mais se alinham ao objetivo deste trabalho. Desse modo, o conjunto de fotos utilizado foi de 83 imagens.

É possível observar que o método funciona com fidelidade até mesmo em fotos onde o paciente está com os olhos entreabertos (Figura 16) e também em pacientes com estrabismo, nos quais a íris não se encontra no centro do olho, casos que costumam apresentar dificuldades na detecção por outros métodos.

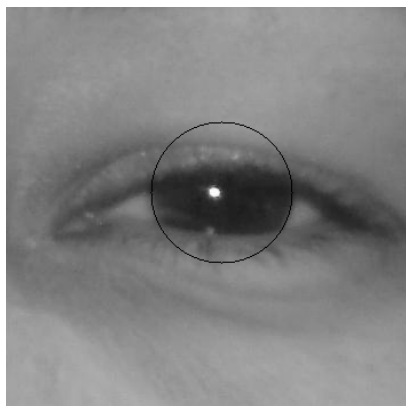


Figure 16: Detecção do Limbo em olho entreaberto.

A base de fotos utilizada inclui máscaras com a marcação manual da posição e tamanho dos limbos. Foram calculadas algumas métricas de desempenho (Subseção 2.7) do método proposto, analisando as imagens resultantes da detecção em relação às máscaras da base, provenientes de marcação manual do limbo. Como comparação, também foram calculadas as métricas de desempenho do detector de círculo de Hough nativo do OpenCV. Os valores encontrados são exibidos na Tabela 2.

Observa-se que o desempenho do Detector de Círculos de Hough implementado neste trabalho é superior em todas as métricas testadas em relação ao Detector de Círculos de Hough nativo do OpenCV Python. No entanto, o tempo médio de execução do Hough nativo do OpenCV foi de 1,45 segundo por imagem. Já o Detector implementado neste trabalho e modificado para buscar especificamente por setores circulares, teve um tempo médio de execução foi

de aproximadamente 41,3 segundos. Os tempos apresentados representam a duração completa do processo de detecção do limbo para ambos os detectores, com o tempo de pré-processamento (recorte da área dos olhos, tratamento da imagem e a extração das bordas) já incluso. Os tempos citados se referem ao processamento em uma máquina com processador Intel i3 3217U, 1.8 GHz, 4 núcleos, e 4 Gb de memória RAM, o código implementado na linguagem Python 3.6, e apenas um núcleo do processador sendo utilizado por vez, devido a um mecanismo de segurança chamado GIL (*Global Interpreter Lock*), que é utilizado pelo interpretador do Python.

#### 4.2 Medição do limbo em réplicas de olhos

O método proposto para cálculo do tamanho do limbo em milímetros foi aplicado em um ambiente controlado para testar o seu funcionamento. Foi elaborado um experimento simples onde fotografias de réplicas de olhos foram retiradas, com o dispositivo e o objeto fotografado sendo posicionados estaticamente em diferentes distâncias desejadas, assim como a iluminação do local, podendo ser regulada conforme a necessidade. O experimento pode ser melhor compreendido observando a Figura 17.

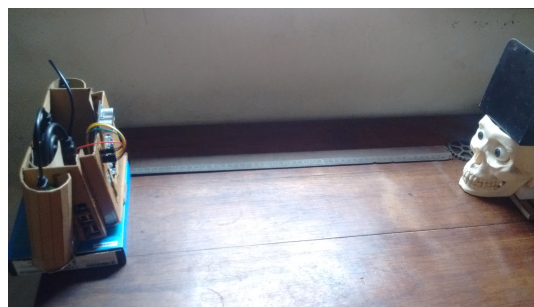


Figure 17: Experimento em ambiente controlado.

As distâncias utilizadas foram de 20, 25 e 30 cm do objeto fotografado. As distâncias utilizadas no cálculo foram o resultado da média de 5 medidas consecutivas realizadas após a foto, prática recomendada em atividades laboratoriais que busca reduzir a influência de possíveis erros em cada medição individual.

Na Figura 18 é exibido um exemplo de como a região equivalente ao limbo na réplica de olho foi detectada pelo método proposto, como esperado. Na Tabela 3 são exibidos os resultados do cálculo do diâmetro do limbo em milímetros com base no diâmetro em pixels do limbo detectado na imagem e a distância entre a câmera e os olhos.

Os modelos de olhos têm o diâmetro do limbo de 12 mm, diâmetro aferido manualmente com um paquímetro<sup>4</sup>. Sendo o erro de cada resultado a diferença entre ele e o valor que deveria ser (12 mm), o erro médio dos resultados é de aproximadamente 0,2 mm. Na prática médica, variações abaixo de 0,5 mm normalmente são desprezadas, sendo o valor medido arredondado para o múltiplo de 0,5 mm mais próximo. Portanto, a metodologia proposta para o cálculo se demonstra compatível com a medida manual, uma vez que ao arredondarmos, o múltiplo mais próximo é o de 12 mm.

#### 4.3 Medição do limbo em pacientes reais

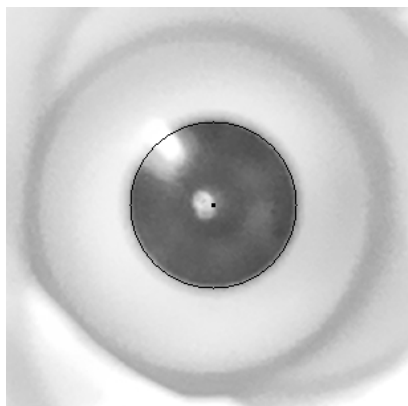
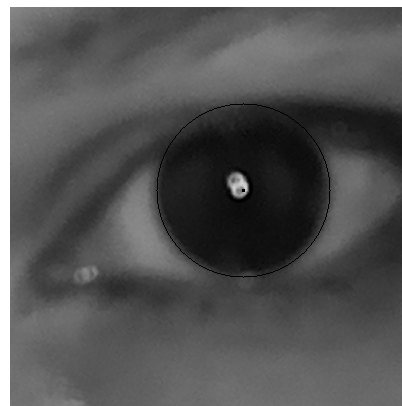
<sup>4</sup>Instrumento empregado para medir precisamente pequenas distâncias, espessuras etc.

**Table 2: Métricas do Hough do OpenCV e do Hough modificado**

Métrica	Hough OpenCV (%)	Hough Modificado (%)
Sensibilidade	88,39	93,97
Especificidade	99,75	99,92
Precisão	82,22	90,76
Acurácia	99,67	99,88
Coefficiente de Dice	83,03	91,89

**Table 3: Resultados do método de detecção do limbo nas réplicas de olhos**

Foto	Dist. (mm)	OD (px)	OE (px)	OD (mm)	OE (mm)
1	205,5	184	184	11,9	11,9
2	254,4	154	152	12,4	12,2
3	302,8	126	126	12	12
4	203,3	184	184	11,8	11,8
5	256	148	152	12	12,3
6	304,6	126	128	12,1	12,3
7	209,1	188	184	12,4	12,2
8	209,4	188	184	12,4	12,2
9	253	152	152	12,1	12,1
10	257,9	152	152	12,4	12,4
11	302,6	128	128	12,2	12,2
12	301,5	128	128	12,2	12,2
Média	-	-	-	12,1583	12,15

**Figure 18: Detecção do limbo na réplica de olho.****Figure 19: Detecção do limbo em imagem de paciente real.**

Com a permissão do médico e dos pacientes, foram capturadas imagens de 5 pacientes em período pré-operatório utilizando o dispositivo com sensor de distância. O médico costuma aferir manualmente o diâmetro do limbo durante a cirurgia, então foi solicitado que posteriormente informasse tais valores para compor a base de dados. O método de detecção do limbo e o cálculo do diâmetro limbar em milímetros foram aplicados à base de imagens retiradas dos pacientes. Na Figura 19 é mostrado um exemplo da detecção do limbo de um dos pacientes.

Na Tabela 4 são apresentadas as medidas dos diâmetros encontrados manualmente pelo especialista, os resultados encontrados automaticamente pelo método proposto neste trabalho, assim como a diferença em módulo entre as duas medidas, caracterizando o erro, para os olhos direito (OD) e esquerdo (OE) respectivamente em cada coluna. Os valores

encontrados foram arredondados para o múltiplo de 0,5 mm mais próximo.

Apesar da resolução da câmera do dispositivo ser alta, as imagens não ficaram nítidas, isto pode ter sido causado devido à falta de foco ajustável na câmera e a pequenos movimentos durante a realização das fotos, tanto do paciente quanto do dispositivo por ser segurado nas mãos e não em uma plataforma fixa. Tais movimentos interferem na medida do sensor de distância, e a falta de nitidez nas imagens ocasionam imprecisões na detecção do limbo. Os pequenos erros nos dados de entrada se propagam ao longo dos cálculos gerando diferenças no resultado final, em algumas aplicações, tais diferenças são desprezíveis, mas para a aplicação deste trabalho, que busca alta precisão, variações acima de 0,5 mm devem ser consideradas.

Table 4: Resultados do método de detecção do limbo em pacientes reais

Paciente	OD e OE reais (mm)	OD e OE calculados (mm)	Erro (mm)
1	12,0 12,0	14,0 14,0	2,0 2,0
1	12,0 12,0	13,5 13,5	1,5 1,5
1	12,0 12,0	13,0 13,0	1,0 1,0
1	12,0 12,0	13,0 13,0	1,0 1,0
2	11,0 11,0	12,5 12,5	1,5 1,5
2	11,0 11,0	12,5 12,5	1,5 1,5
3	11,5 11,5	12,5 12,5	1,0 1,0
3	11,5 11,5	12,5 12,5	1,0 1,0
3	11,5 11,5	13,0 13,0	1,5 1,5
3	11,5 11,5	13,5 13,5	2,0 2,0
4	11,5 11,5	11,0 11,0	0,5 0,5
4	11,5 11,5	11,0 11,0	0,5 0,5
4	11,5 11,5	12,0 12,0	0,5 0,5
4	11,5 11,5	12,0 12,0	0,5 0,5
4	11,5 11,5	11,0 11,0	0,5 0,5
5	11,0 11,0	12,5 12,5	1,5 1,5
5	11,0 11,0	11,5 11,5	0,5 0,5
5	11,0 11,0	14,0 13,0	3,0 2,0
5	11,0 11,0	13,0 13,0	2,0 2,0
5	11,0 11,0	13,0 13,0	2,0 2,0
Média	-	-	1,275 1,225

## 5. CONCLUSÕES

Problemas visuais podem diminuir a capacidade visual de pacientes, e em alguns casos, levar até à cegueira. O diâmetro limbar é uma medida utilizada pelos oftalmologistas em alguns exames relacionados à saúde dos olhos de pacientes, como os que possibilitam a detecção do glaucoma congênito, problema que atinge crianças em idade pré-escolar e escolar e pode prejudicar o desempenho delas na escola, o desenvolvimento motor e a capacidade de comunicação do indivíduo em formação. Desta forma, o diagnóstico precoce de patologias como esta é importante para que o tratamento seja feito o quanto antes.

Existem dificuldades em aplicar os exames usuais em crianças, pois geralmente a colaboração do paciente se faz necessária, e crianças pequenas não conseguem colaborar. A prática médica atual consiste em sedar o paciente e utilizar de técnicas invasivas e nem sempre precisas para o exame. Assim, se faz oportuna uma solução computacional que evite tais problemas, permitindo que se faça a detecção automática do limbo em imagens e que seu diâmetro seja calculado em milímetros.

O uso de processamento de imagens para detecção dos olhos e localização do limbo estão presentes na literatura, porém, são utilizados em aplicações como exames de estrabismo, segmentação da pupila para testes do reflexo vermelho, rastreamento da direção dos olhos em vídeo e reconhecimento biométrico por meio da íris. Neste trabalho, as etapas de detecção de olhos, localização do limbo e cálculo do diâmetro do limbo em milímetros foram implementadas na linguagem Python 3.6, com auxílio da ferramenta OpenCV e do dispositivo proposto por [21], com algumas modificações de hardware e software para integrar novos componentes, como o sensor de distância.

Inicialmente planejou-se que o processamento fosse realizado na placa *Raspberry Pi* do dispositivo, no entanto, devido a alterações nas técnicas utilizadas no método proposto,

em busca de melhores resultados, o baixo poder de processamento e memória do *Raspberry* se tornaram fatores limitantes, por isso, o dispositivo passou a ser usado apenas para captação da imagem e distâncias, os dados registrados passaram a ser armazenados em arquivos e exportados para um computador tradicional, onde foi realizado o processamento dos dados.

O método desenvolvido demonstrou funcionar de forma coerente com o que foi proposto. Conseguiu-se realizar medições do diâmetro limbar em milímetros com base em processamento de imagens em conjunto com outros dados, como especificações da câmera e a distância entre a câmera e o paciente. Os resultados obtidos através do método desenvolvido são compatíveis com os valores obtidos pelo método manual de medição.

Em alguns casos, a precisão dos resultados não foi tão alta quanto desejada, devido a fatores como o foco não ajustável da câmera utilizada, a irregularidade de iluminação em ambientes não controlados e a natureza irregular da superfície do rosto humano, que interfere no correto funcionamento de sensores de distância baseados em ultrassom, o tipo que foi utilizado. Apesar disso, o desempenho obtido pela técnica desenvolvida indica que, com equipamentos diferentes, de melhor precisão, a metodologia proposta tem potencial de aplicação em consultórios oftalmológicos.

Possíveis aperfeiçoamentos que podem ser feitos em trabalhos futuros são: melhoria da câmera, optando-se por uma com foco ajustável ou, preferencialmente, autoajustável, uso de sensor de distância baseado em infravermelho ou laser, pois possuem melhor precisão na medida, uso de uma versão superior da placa *Raspberry Pi*, com maior capacidade de processamento, projetar um *case* para o dispositivo que possa ser impresso em uma impressora 3D, adaptado para o encaixe do *Raspberry*, do *display*, das baterias e da placa de circuito com a câmera e os LEDs de iluminação, implementação do método em outra linguagem de programação que

possua execução mais rápida e com menos uso de memória, como C, e estudar a possibilidade do uso de *threads* para processamento paralelo, em busca da redução do tempo processamento do método.

Em um contexto mais amplo, este trabalho se integra a um projeto de desenvolvimento de uma plataforma *open-source* e *open-hardware*, na qual os códigos dos métodos computacionais para detecção de patologias da visão seriam disponibilizados, assim como detalhes do hardware utilizado e as instruções de montagem, de modo que profissionais da área da saúde possam adquirir o hardware e incluir nele os diversos métodos, facilitando assim o acesso de mais pessoas a exames e processos de triagem, tornando mais eficiente o funcionamento de sistemas de saúde.

## 6. AGRADECIMENTOS

Os autores agradecem ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPQ) - Processo 307210/2018-9, pelo apoio financeiro.

## 7. REFERENCES

- [1] J. Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.
- [2] P. Chakravorty. What is a signal?[lecture notes]. *IEEE Signal Processing Magazine*, 35(5):175–177, 2018.
- [3] A. Couto Junior and L. A. G. d. Oliveira. As principais causas de cegueira e baixa visão em escola para deficientes visuais. *Revista Brasileira de Oftalmologia*, 75(1):26–29, 2016.
- [4] F. R. G. Cruz, C. C. Hortinela, B. E. Redosendo, B. K. P. Asuncion, C. J. S. Leoncio, N. B. Linsangan, and W.-Y. Chung. Iris recognition using daugman algorithm on raspberry pi. In *2016 IEEE Region 10 Conference (TENCON)*, pages 2126–2129. IEEE, 2016.
- [5] J. D. S. De Almeida, A. C. Silva, J. A. M. Teixeira, A. C. Paiva, and M. Gattass. Computer-aided methodology for syndromic strabismus diagnosis. *Journal of digital imaging*, 28(4):462–473, 2015.
- [6] J. EHLERS and C. SHAH. Manual de doenças oculares do wills eye hospital: diagnóstico e tratamento no consultório e na emergência. *Porto Alegre: Artmed*, 2009.
- [7] Dr. A. Frazão. Glaucoma: quais os exames que confirmam o diagnóstico. 2016.
- [8] R. C. Gonzalez and R. C. Woods. *Processamento digital de imagens*. Pearson Educación, 2009.
- [9] A. Haar. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 71(1):38–53, 1911.
- [10] P. V. Hough. Method and means for recognizing complex patterns, Dec. 18 1962. US Patent 3,069,654.
- [11] N. E. Institute. Glaucoma, 2019.
- [12] Z. Kunik, A. Bykowski, T. Marciniak, and A. Dąbrowski. Raspberry pi based complete embedded system for iris recognition. In *2017 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, pages 263–268. IEEE, 2017.
- [13] R. Laganière. *OpenCV Computer Vision Application Programming Cookbook Second Edition*. Packt Publishing Ltd, 2014.
- [14] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *Proceedings. international conference on image processing*, volume 1, pages I–I. IEEE, 2002.
- [15] R. M. Lorenzi. *Software de Detecção de Estrabismo para Dispositivos Móveis*. Monografia, Universidade Regional de Blumenau, Curso de Ciência da Computação, Blumenau, 2014.
- [16] OpenCV. Smoothing images, 2019.
- [17] N. Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.
- [18] C. P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*, pages 555–562. IEEE, 1998.
- [19] L. G. Shapiro and G. Linda. stockman, george c. *Computer Vision*, Prentice hall, 2002.
- [20] I. F. S. d. SILVA et al. Detecção automática da presença de patologia na visão baseada em imagens do teste de brückner. 2019.
- [21] R. M. Silveira, J. D. Almeida, J. A. Teixeira, I. M. Maia, A. C. Paiva, and G. B. Júnior. Dispositivo de baixo custo para detecção de patologias da visão. In *Anais do XVIII Simpósio Brasileiro de Computação Aplicada à Saúde*. SBC, 2018.
- [22] I. Sobel and G. Feldman. A 3x3 isotropic gradient operator for image processing. *a talk at the Stanford Artificial Project in*, pages 271–272, 1968.
- [23] S. E. Umbaugh. *Digital image processing and analysis: human and computer vision applications with CVIPtools*. CRC press, 2010.
- [24] P. Viola, M. Jones, et al. Rapid object detection using a boosted cascade of simple features. *CVPR (1)*, 1(511-518):3, 2001.
- [25] H. Zhang, K. Wiklund, and M. Andersson. Circle detection using isosceles triangles sampling. *arXiv preprint arXiv:1511.00461*, 2015.