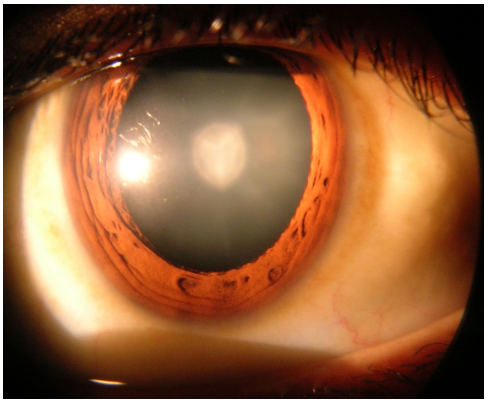
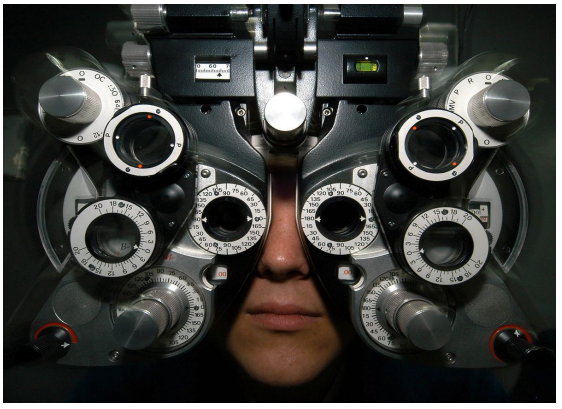
Introdução a Técnicas de Programação - 2017.2

Descrição de projeto

Diagnóstico de Catarata





Catarata é uma opacificação do cristalino do olho que causa diminuição da capacidade visual. Pode afetar um ou ambos os olhos e é frequente desenvolver-se lentamente. Os sintomas podem incluir visão desfocada, diminuição de sensibilidade às cores, halos à volta das luzes, dificuldade em observar luzes brilhantes e dificuldade em ver durante a noite. Isto poderá afetar a condução, leitura, ou reconhecimento de rostos. A diminuição da capacidade visual pode também aumentar o risco de acidentes e depressão. As cataratas são a causa de metade dos casos de cegueira e de um terço dos casos de incapacidade visual em todo o mundo.

As cataratas são formadas por depósitos de proteínas ou pigmentos amarelados no cristalino, que diminuem a transmissão de luz para a retina na parte de trás do olho. A maior parte dos casos de cataratas deve-se ao envelhecimento da pessoa, mas a doença pode também ter origem em traumas ou exposição à radiação, estar presente desde o nascimento ou ocorrer na sequência de uma cirurgia ocular ou de outros problemas. Entre os fatores de risco estão a diabetes, fumar, a exposição prolongada à luz do sol e o consumo de bebidas alcoólicas. O diagnóstico é realizado através de um exame ocular.

As medidas de prevenção incluem o uso de óculos de sol e deixar de fumar. Na fase inicial os sintomas podem ser aliviados com o uso de óculos. Quando os óculos não resultam, o único tratamento eficaz é uma cirurgia para remover o cristalino opaco e substituí-lo por uma lente artificial. A cirurgia só é necessária nos casos em que as cataratas causam problemas. A cirurgia geralmente melhora a qualidade de vida. No entanto, em muitos países não é facilmente acessível, principalmente para mulheres, para pessoas que vivem no meio rural e para pessoas que não conseguem ler.

Em todo o mundo há cerca de 20 milhões de pessoas com cegueira provocada por cataratas. A doença é a causa de 5% dos casos de cegueira nos Estados Unidos e de cerca de 60% em partes de África e da América do Sul. A cegueira causada por cataratas afeta entre 10 a 40 em cada 100 000 crianças nos países em vias de desenvolvimento e entre 1 a

4 em cada 100 000 nos países desenvolvidos. As cataratas tornam-se mais comuns com a idade. (Wikipedia)

O Embasamento teórico do Projeto

Nesse projeto você deve fazer um programa em C para analisar 4 imagens (duas com catarata e duas sem catarata) e emitir o diagnóstico final de cada. Para isso é preciso realizar algumas etapas:

1. **Transformação da imagem colorida para tons de cinza**

Para converter qualquer cor em seu nível aproximado de cinza, deve-se primeiro obter suas primitivas vermelho, verde e azul (da escala RedGreenBlue). Adiciona-se então 30% do vermelho mais 59% do verde mais 11% do azul, independente da escala utilizada (0.0 a 1.0, 0 a 255, 0% a 100%.) O nível resultante é o valor de cinza desejado. Tais porcentagens estão relacionadas à própria sensibilidade visual do olho humano convencional para as cores primárias.

2. **Segmentação da imagem**

A **filtragem** é um procedimento usual ao realizar o tratamento de uma imagem, antes de que um determinado processamento possa ser feito. De uma forma geral a filtragem de imagens é implementada através de uma *multiplicação especial* entre duas matrizes (operação também chamada de convolução).

$$Ires(x,y) = \sum_{a=-M/2}^{M/2} \sum_{b=-N/2}^{N/2} F(a+M/2, b\ +\ N/2)\ Iorig(x+a, y+b)$$

Na Equação 1, ***Ires e Iorig*** representam, respectivamente, a imagem resultante após a filtragem e a imagem original a ser filtrada; ***Ires(x,y) e Iorig(x,y)*** representam os pixels das imagens nas posições x e y de cada uma. Os valores de M e N são os tamanhos da matriz F, que é o filtro a ser aplicado. Existem diversas utilizações para filtragem tais como remoção ou adição de ruído na imagem e mudança ou realce de algumas características. Neste trabalho iremos aplicar dois filtros na imagem: o Filtro Gaussiano e o Filtro de Realce de Arestas.

O **filtro gaussiano** é um dos mais comuns em processamento digital de imagens, pois ele reduz as transições bruscas na imagem (ruídos de alta frequência), deixando a imagem mais “suave”. Na prática o filtro gaussiano é implementado através da operação de filtragem especificada na Equação 1 usando uma matriz F, obtida através de uma função gaussiana 2D. Para o nosso caso, usaremos os valores para F como mostrados na Figura 1, abaixo:

$$\frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

Figura 1, Matriz de um possível Filtro Gaussiano

Realce de Arestas é um processo de realce de arestas na imagem. Esse processo recebe como entrada uma imagem e produz como saída outra imagem, porém contendo apenas os contornos da imagem original, em preto e branco. Para muitos procedimentos que precisam da detecção de figuras geométricas, como círculos, no nosso caso, é necessário o uso de imagens que contenham apenas os contornos das figuras na imagem ao invés dos valores de todos os pixels em RGB. Para tal é necessário realizar uma série de procedimentos a fim de garantir a detecção dos contornos com melhor qualidade (detector de arestas Canny):

a. Transformação da imagem para escalas de cinza
b. Aplicação de um filtro gaussiano
c. Filtragem com sobel
d. Cálculo das magnitudes em x e y, e cálculo das angulações das arestas
e. Atenuação/remoção dos valores que não são arestas
    i. Processamento de vizinhança na direção do gradiente
    ii. Remoção de valores através de duplo threshold

A **Binarização** é um tipo de processamento que se baseia na diferença dos níveis de cinza que compõe diferentes objetos de uma imagem. A partir de um limiar estabelecido de acordo com as características dos objetos que se quer isolar, a imagem pode ser segmentada em dois grupos: o grupo de pixels com níveis de cinza abaixo do limiar e o grupo de pixels com níveis de cinza acima do limiar. Em uma imagem limiarizada, atribui-se um valor fixo para todos os pixels de mesmo grupo. No nosso caso, aplicaremos um procedimento de binarização na imagem gerada a partir do realce de arestas. A imagem resultante após o processo de realce de arestas ainda contém valores de pixel em tons de cinza, variando entre 0 e 255 dependendo do quão “forte” é a aresta em questão. O procedimento de binarização irá transformar essa imagem em uma imagem binária (com valores de 0 e 1, ou valores mínimos e máximos), onde os valores máximos representarão pixels que, com certeza, estão em alguma aresta da imagem, enquanto os valores mínimos representarão pixels que não estão em aresta alguma e, portanto, devem ser desconsiderados.

A **Transformada de Hough** é uma técnica matemática que realiza a detecção de formas geométricas em imagens digitais. Em sua forma original a transformada de Hough foi elaborada por Paul Hough em 1962. Sua primeira concepção estava baseada na localização de retas. Posteriormente, a transformada de Hough foi estendida para possibilitar a localização de outras formas geométricas que possam

ser parametrizadas, tais como círculos e elipses. A transformação de Hough para detecção de círculos processa a imagem dada como entrada e produz como saída uma representação em um espaço de quatro dimensões, contendo valores possíveis de todos os centros de círculos com determinado raio que podem interseccionar um dado pixel da imagem. O procedimento para produzir a representação no espaço de Hough para uma imagem é a seguinte:

a. Transforme a imagem em uma representação de arestas
b. Processe todos os pixels da imagem de arestas de acordo com o algoritmo abaixo:

```
para cada pixel(x,y) que pertence a uma aresta
  // os raios possíveis devem ser conhecidos, é necessário fazer
  // ajustes manuais nessa parte para detecção de círculos
  // com raios  maiores que rmax pixels ou menores que rmin pixels
 para cada raio r = rmin à r = rmax
    para cada theta t = 0 à 360 // valores padrão
      a = x – r * cos(t * PI / 180); //coordenada a do centro
      b = y – r * sin(t * PI / 180);  //coordenada b do centro
      A[a,b,r] +=1; //espaço de Hugh
    end
  end
end
```

c. Ao fim do algoritmo a matriz A[a,b,r] contém todos os centros (a,b) para todos os círculos de raios (r), no intervalo de raios considerados, que interseccionam todos os pixels que pertencem às arestas da imagem. Assim, se uma imagem contém apenas UM círculo, com centro (Ac, Bc) e raio Rc (com rmin<=Rc<=rmax); espera-se que o valor de A[Ac,Bc,Rc] seja maior do que outros valores da matriz A, indicando assim a presença do círculo na posição e raios especificados na matriz.
d. Caso a imagem contenha mais de um círculo, é necessário encontrar mais de um máximo na matriz A, por exemplo, podemos, ao invés de procurar apenas pelo maior valor de A, buscar pelos n maiores valores, que representam as n circunferências detectadas pelo algoritmo.

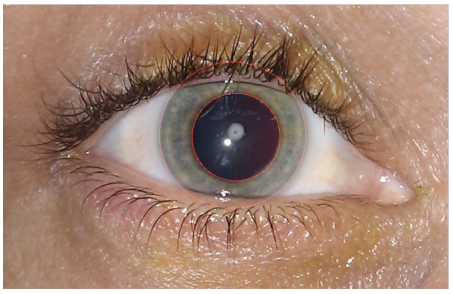


Figura1. Encontrando a pupila

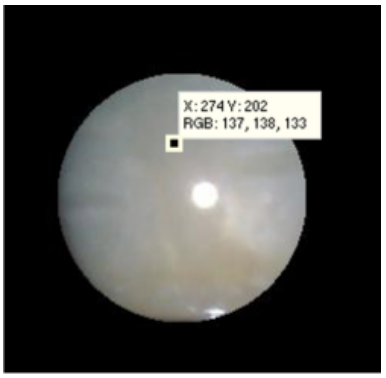


Figura 2. Pupila Segmentada e valores RGB de um pixel específico.

3. Classificação

Analisar cada pixel da pupila e determinar se é um pixel de catarata ou não de acordo com um limiar (*threshold = 60%-70%*). Fazer a contagem de todos os pixels de catarata presentes na pupila e analisar a quantidade dos mesmos em relação a quantidade de pixel total, resultando em uma porcentagem de comprometimento da pupila.

Requisitos funcionais do programa

O nome do binário/executável deverá ser "**catarata**" e o mesmo deverá ser capaz de processar um conjunto de parâmetros, como indicado a seguir:

-i <input-image> a imagem de entrada a ser processada
-f <input-image-format> indica o formato da imagem de entrada
-o <diagnose-file> indica o nome do arquivo texto contendo o diagnóstico

Exemplo de execução: ./catarata -i imagem.bmp -f bmp -o diagnostico.txt

O arquivo de diagnóstico deve conter:

a) Diagnóstico Geral: Com/Sem catarata
b) Porcentagem de Comprometimento: <porcentagem>%.

Ps: observar que cada etapa do trabalho (check point) tem uma saída específica que será cobrada.

Por questões de simplicidade, seu programa deverá suportar OBRIGATORIAMENTE o formato de imagem PPM (Portable PixMap): imagens a cores em formato decimal (P3).

Formato de imagem

Consultar o documento "Formatos de Imagem", texto elaborado por João Manuel Brisson Lopes para a disciplina Computação Gráfica do curso Licenciatura em Engenharia Informática e de Computadores no Departamento de Engenharia Informática do Instituto Superior Técnico - Universidade Técnica de Lisboa. Este documento foi publicado em Janeiro de 2003 e reeditado em Dezembro de 2008 e Abril 2013. Disponível em: http://disciplinas.ist.utl.pt/~leic-cg.daemon/textos/livro/Formatos%20de%20Imagem.pdf

Descrição do projeto

O projeto de diagnóstico de catarata deve ser desenvolvido em linguagem C, a ser executado, em sua versão mais simples, através de linha de comando (entrada e saída em um console/terminal). O projeto deve atender os seguintes critérios de programação:

1. **Alocação dinâmica de arranjos e/ou matrizes;**
2. **Uso de registros (`struct`) e enum para representar/ler/gravar uma imagem;**
3. **Definição de novos tipos de dados através de `typedef`;**
4. **Leitura/escrita de imagens a partir de arquivos;**
5. **Modularização do programa em diferentes funções (modularização interna) e arquivos (uso de diferentes arquivos .c e .h, cada um com sua funcionalidade - modularização externa);**
6. **Boas práticas de programação: Definição de um padrão de indentação do código fonte e de nomenclatura das sub-rotinas e variáveis;**
7. **Emitir mensagens para stderr em situações de erro: falha na alocação de memória, falha na abertura do arquivo.**
8. **Documentação adequada do código-fonte (uso de comentários).**

Observações:

- O projeto deve ser desenvolvido **individualmente ou em duplas** (grupos de dois alunos). Não serão permitidos grupos com três ou mais alunos;
- Para facilitar o acompanhamento do projeto, cada dupla deve estar associada a uma única turma de PTP e a uma única turma de ITP. Ou seja, não serão permitidas duplas formadas por alunos matriculados em turmas diferentes;
- Cada dupla deve desenvolver sua solução de forma independente das demais. Soluções idênticas serão consideradas plágios e, portanto, sanções serão devidamente aplicadas em todas as duplas com soluções similares;

- Códigos e algoritmos podem ser utilizados da web desde que devidamente referenciados. Caso sejam encontrados trechos de código na web equivalentes aos apresentados pelo grupo sem a devida citação, o código será igualmente considerado plágio e sanções serão aplicadas ao grupo. Vale salientar que a avaliação será realizada unicamente sobre o código produzido pela dupla. Códigos retirados da web, apesar de permitidos com a devida citação, serão desconsiderados dos critérios de pontuação.

Processo de Avaliação

O processo de avaliação desta tarefa compreende a execução e desenvolvimento do projeto em **4 semanas**. A cada semana, o grupo deve apresentar uma etapa do projeto desenvolvido, seguindo o calendário abaixo:

- **CHECKPOINT 1 - 13/11/2017**
  - **Leitura da Imagem**
    - **Especificações:**
    - Tipos de dados necessários (`typedef`, `structs` e `enums`) para manipular cada formato de imagem a ser suportado;
    - Modularização externa do programa (quais os arquivos .c e .h);
    - Leitura de imagens PPM de arquivo e correta leitura dos pixels;
    - **Entradas:**
    - Imagens em Formato em PPM
    - **Saídas:**
    - N/A
- **CHECKPOINT 2 - 20/11/2017**
  - **Transformação em tons de cinza**
  - Entradas
    - Imagens em formato PPM
  - Saídas
    - Imagens em formato PPM em tons de cinza
- **CHECKPOINT 3 - 22/11/2017**
  - **Segmentação das imagens em tons de cinza**
  - Entradas
    - Imagens em formato PPM
  - Saídas
    - Imagem com a **pupila** segmentada em formato PPM em tons de cinza
  - Saída opcional (extra)
    - Imagem RGB com contorno da pupila
- **CHECKPOINT 4 - 27/11/2017**
  - **Detecção da Catarata**
  - Entradas
    - Imagens em formato PPM em tons de cinza

- Saídas
    - Diagnóstico e porcentagem de comprometimento da pupila

❑ **CHECKPOINT "Atrasados é a sua última chance!" - 29/11/2017**

1. Funcionalidades adicionais.
2. O que falta apresentar.

Sobre as duplas

Os alunos têm **ATÉ o dia 01/11/2017** para comunicar aos respectivos professores de ITP e PTP **(SOMENTE POR E-MAIL)** se farão o trabalho em dupla (e a composição da mesma) ou se farão individualmente. O assunto do e-mail deve ser: "**Trabalho IMD0012 - 2017.2: Definição dupla**" e no corpo do e-mail deve conter os nomes completos da dupla.

Critérios de pontuação

O desenvolvimento do projeto aqui descrito vale 100% da nota da terceira unidade de PTP e ITP. A pontuação da avaliação seguirá os critérios e distribuição abaixo:

❑ Atendimento dos requisitos funcionais: 50%

❑ Uso dos recursos da linguagem C: 20%

A dupla demonstrou saber usar de forma adequada os recursos da linguagem C (arranjos, structs, typedefs, recursividade, etc)?

❑ Organização do código e documentação: 10%

O código está documentado? a indentação e uso de { } seguem um padrão (indentação)? o programa está devidamente modularizado em diferentes arquivos?

❑ Funcionalidades adicionais: 20%

As funcionalidades adicionais desenvolvidas pela dupla foram suficientemente complexas?

A pontuação a ser dada pelas funcionalidades extras não é definida *a priori*. Cada caso será avaliado em função da complexidade envolvida. Itens extras de baixa complexidade serão desconsiderados na pontuação.

Entrega do projeto

O projeto deve ser submetido pelo SIGAA até a data **01/12/2017 (até o final do horário da respectiva aula)** em um arquivo comprimido **(.zip)** contendo os arquivos fontes do projeto **(.c e .h)** e um arquivo **README.TXT**. Este arquivo deve ter as seguintes informações:

❑ O que foi feito (o básico e as funcionalidades extras implementadas);

❑ O que não foi feito (caso não tenha sido possível implementar alguma funcionalidade);

- ❑ O que seria feito diferentemente (quando aprendemos à medida que vamos implementando, por vezes vemos que podíamos ter implementado algo de forma diferente. Assim, esse tópico é para vocês refletirem sobre o que vocês aprenderam durante o processo que, se fossem fazer o mesmo projeto novamente, fariam diferente);
- ❑ Como compilar o projeto. Atenção, não será permitido o uso de bibliotecas externas. Vocês devem usar APENAS a biblioteca padrão de C que foram vistas em sala de aula (stdio, math, stdlib, string).
- ❑ Em caso de duplas:
    - Identificação dos autores;
    - Contribuição de cada integrante no desenvolvimento do projeto (quem fez o quê).

  A presença desse dia será a entrega do projeto. Não haverá apresentação.

Recomendações diversas:

1. Solução de backup: não será tolerada a desculpa de que um disco rígido falhou nas avaliações. Assim, é importante manter várias cópias do código-fonte desenvolvido ou usar um sistema de backup automático como o Dropbox, Google Drive, Box ou similares. Uma solução melhor ainda é fazer uso de um sistema de controle de versões como git, e um repositório externo como Github ou Bitbucket.
2. Especificar precisamente a interface e o comportamento de cada sub-rotina. Usar esta informação para guiar o desenvolvimento e documentar o código desenvolvido.

Adicionais

1. Implementação de outras técnicas de identificação de catarata.
2. Detecção de catarata nas imagens extras
3. Segmentação do flash nas imagens extras