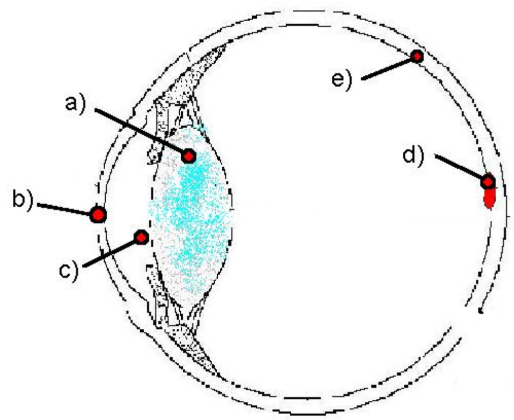
MAC 417 – Visão e Processamento de Imagens - Parte I

Resolução da Primeira Prova - 27 de setembro de 2004

Problema 1: [2.5 pontos] Nomeie as partes da estrutura do olho indicadas com (a), (b), (c), (d), (e) e descreva a função de cada uma.



A córnea (b) é uma membrana transparente na parte externa do olho, que cobre o humor aquoso e íris, sendo a primeira superfície do olho atravessada pela luz que entra no olho. A parede externa do olho (e) é formada pela esclera, que é a parte branca mais externa do olho, a a coróide, que é uma membrana abaixo da esclera que possui uma rede de vazos sanguíneos e é altamente pigmentada para impedir a passagem de luz, e a retina, que contém as células fotosensíveis (entre outras). O cristalino ou lente (a) é responsavél pela focalização da luz que entra pela pupila (c) sobre a fóvea (d). A pupila é um buraco no meio da íris que controla a quantidade de luz que entra no olho, e a fóvea é uma região da retina que contém uma grande concentração de cones, e é responsável pela nossa visão fotópica.

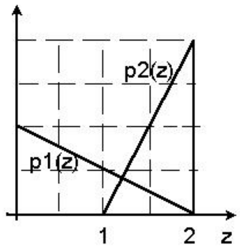
Problema 2: [2.5 pontos] Forneça um algoritmo que transforme uma imagem binária, com bordas de largura 1 pixel com conectividade-m, em uma imagem com bordas de conectividade-4. Seu algoritmo deve pintar o mínimo número de pixels possível.

O problema de se pintar o menor número possível exige alguns cuidados. O algoritmo a seguir está bem mais detalhado que o necessário.

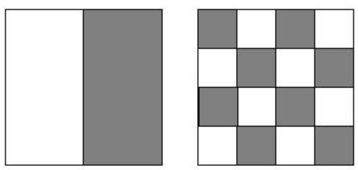
```
Vamos adotar os seguintes nomes para os vizinhos do pixel x
  NO   N   NL
  O    x    L
  SO   S   SL

#define DIAGONAL 0
#define NORTE 1
#define SUL 2
#define LESTE 3
#define OESTE 4
#define Borda valor_de_borda

Pseudo Algoritmo, considerando uma lista L de pixels borda:
Para todos os pixels em L:
   switch (NumViz8(x)) {
```

```
      case 4: // igual case 3
      case 3: O(x) = Borda; L(x) = Borda; break;  // ou N S
      case 2: switch (QueVizinhos8(x)) {
                   case DIAGONAL: O(x) = Borda; L(x) = Borda; break;
                   case NORTE: N(x) = Borda; break;
                   case SUL  : S(x) = Borda; break;
                   case LESTE: L(x) = Borda; break;
                   case OESTE: O(x) = Borda; break;
              }
      case 1: if ((NO(x) == Borda) OU (SO(x) == Borda)) O(x) = Borda;
              else L(x) = Borda;
      default: // do nothing -> já é vizinho 4
   }

Rotinas:
N(x), S(x) ..., SO(x) referem ao pixel Norte de x, sul de x, etc.

int NumViz8 (x) { // retorna o número de vizinhos 8 do pixel x
  int cont= 0;
  if (NO(x) == Borda) AND (N(x) != Borda) AND (O(x) != Borda) cont++;
  if (SO(x) == Borda) AND (S(x) != Borda) AND (O(x) != Borda) cont++;
  if (NL(x) == Borda) AND (N(x) != Borda) AND (L(x) != Borda) cont++;
  if (SL(x) == Borda) AND (S(x) != Borda) AND (L(x) != Borda) cont++;
  return cont;
}

int QueVizinhos8(x) { // já sei que NumViz8(x) = 2
  int Dno=0, Dnl=0, Dso=0, Dsl=0;
  if (NO(x) == Borda) AND (N(x) != Borda) AND (O(x) != Borda) Dno=1;
  if (SO(x) == Borda) AND (S(x) != Borda) AND (O(x) != Borda) Dso=1;
  if (NL(x) == Borda) AND (N(x) != Borda) AND (L(x) != Borda) Dnl=1;
  if (SL(x) == Borda) AND (S(x) != Borda) AND (L(x) != Borda) Dsl=1;

  if (Dno=Dsl)) return DIAGONAL;
  else if (Dno == Dnl)
     if (Dno == 1) return NORTE;
     else return SUL;
  else // dno é igual a dso
     if (dno == 1) return OESTE;
     else return LESTE;
}
```

Problema 3: [2.5 pontos] Segmentação:
a) [1 pt] As funções densidade de probabilidade de duas regiões de uma imagem em níveis de cinza (que correspondem a objetos), normalizada, é dada na figura abaixo. Nesse caso, qual o limiar ótimo de segmentação assumindo que os objetos são equiprováveis? Justifique.
Como os objetos são equiprováveis, o limiar ótimo deve ser calculado como $P2p1(T) = P1p2(T) => p1(T) = p2(T)$. As retas tem as seguintes equações:
$y = -0.5x + 1$
$y = 2x - 2$
subtraindo as duas equações, temos que x = $3/2.5 = 1.2$, que é o valor do limiar.



b) [0.5 pt] Considere as duas imagens abaixo. O histograma das duas imagens é igual? Justifique.



Sim, o histograma é igual, pois há 2 cores, e a área (número de pixels) com cada uma das cores é a mesma.

c) [1 pt] Considere que a figura acima é convoluida com o filtro abaixo. O histograma das imagens convoluidas é igual? Justifique.

$$\frac{1}{16}\begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline\end{array}$$

Não será igual pois há muito mais pixels de borda na imagem quadriculada que na imagem de barras verticais. Com isso, quando o filtro é aplicado em mais bordas, os pixels nas vizinhanças das bordas são suavizados. Como mais pixels são suavizados na imagem quadriculada, o número de pixels "suaves"será maior, e portanto o histograma é diferente.

Problema 4: [2.5 pontos] Transformada de Hough para linhas:
a) [0.5 pt] Justifique por que ela é interessante computacionalmente para a segmentação de imagens?

Devido ao menor custo computacional. Seja N o número de pixels de borda detectados. Para determinar melhores linhas, pode-se pegar todos os pares de pixel ($O(N^2)$), e para cada um desses pares, determinar quantos dos pixels restantes se alinham com o par, ou seja, testando os demais pixels ($O(N^3)$). O custo computacional do método de Hough é ($O(N)$), pois para cada pixel de borda, varre o espaço paramétrico 1 vez, marcando os possíveis candidatos.

b) [0.5 pt] Mostre como calcular a informação da direção do gradiente para os pixels de borda.

A direção do gradiente pode ser calculada através de máscaras como a de Sobel ou Prewitt, com 2 direções preferenciais, horizontal e vertical. Chamamos a resposta da máscara horizontal dx e a vertical dy. A direção do gradiente é dado por $\tan^{-1}(dy/dx)$.

c) [1.5 pt] Assumindo que você possui a direção do gradiente para cada pixel de borda (mesmo que você não saiba responder o item anterior), descreva, através de um pseudo algoritmo, como reduzir o custo computacional da transformada de Hough para linhas.

Como a direção do gradiente é perpendicular a direção da reta, podemos varrer apenas os ângulos perpendiculares ao gradiente. Para permitir algum erro, considerar uma variação em torno dessa direção.