Universidade Federal Rural de Pernambuco – UFRPE
Curso: Bacharelado em Sistemas de Informação – BSI
Disciplina: Introdução à Programação – Turma SI1 - Prof: Cícero Garrozi

Segundo Projeto – 2015-1– Valor: 10% (1 pt) da 2ª V.A.

Instruções:

- O projeto deve ser feito em grupos de até 3 alunos.
- Data de Entrega: 06/07/2015 até às 23h59m por email. Apresentação pelo grupo: dia 07/07/15 no horário da aula;
- Na data de entrega, enviar os arquivos do projeto compactados (.zip ou .rar) para o email cicerog@gmail.com com o assunto: "[Projeto2-IP-2015-1]". Informar os nomes dos componentes do grupo no conteúdo do email.
- Apresentar o código ao professor em sala, impresso e com o nome dos integrantes, período e nome da disciplina, em 07/07/2015.
- Condições para receber nota 0 (zero):
    1. Entrega fora do prazo estabelecido;
    2. Algoritmos com erros de sintaxe e/ou lógica que prejudiquem o objetivo principal do trabalho;
    3. Algoritmos incompletos;
    4. Algoritmo com alta similaridade com algum de outro(s) grupo(s);
    5. Algum membro não saber explicar o algoritmo;
    6. Não manter a segurança do seu código fonte.

Neste projeto, sua equipe irá desenvolver um algoritmo para transformar uma imagem, exibindo somente o seu contorno (linhas). O processamento da imagem será realizado com o operador de detecção de bordas de Sobel, conforme abordado mais adiante.

Criar um programa em Python que irá receber 2 parâmetros: **o nome do arquivo de entrada** e do **arquivo de saída**, respectivamente. O programa deverá ler o arquivo de entrada (arquivo de imagem no formato PGM – vide informações abaixo), realizar a detecção de bordas através do processamento dos pixels (pontos da imagem) e gravar a imagem resultante no arquivo de saída.

Formato PGM – Portable Gray Map

Este formato armazena as informações de tons de cinza de cada pixel em formato texto. É idêntico ao formato PPM, sendo que armazena apenas tons de cinza. A primeira parte do arquivo é constituída de um cabeçalho que identifica o formato do arquivo, a largura, a altura e o maior valor de um componente de cor do bitmap (quantidade máxima de tons). O identificador do formato é sempre P2 se os valores dos pixels estiverem em formato ASCII ou P5 se estiverem em formato binário. Seu programa deve considerar somente os arquivos no modo texto (P2). O separador de atributos do cabeçalho pode ser o caractere ESPAÇO, TABULAÇÃO ou QUEBRA DE LINHA. O restante do arquivo contem as informações sobre os tons de cinza dos pixels do bitmap, onde cada número equivale a determinada cor (nível de cinza) de um

ponto da imagem (pixel). Qualquer linha iniciada com um caractere # é um comentário e será ignorada.

Exemplo:

```
P2
# feep.pgm
24 7
63
0 0 0 0 0 0 0 0 0 0 0 0  0  0  0  0 0 0  0  0  0  0 0
0 3 3 3 3 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 15 15 15 0
0 3 0 0 0 0 0 7 0 0 0 0 0 11  0  0  0 0 0 15  0  0 15 0
0 3 3 3 0 0 0 7 7 7 0 0 0 11 11 11  0 0 0 15 15 15 15 0
0 3 0 0 0 0 0 7 0 0 0 0 0 11  0  0  0 0 0 15  0  0  0 0
0 3 0 0 0 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15  0  0  0 0
0 0 0 0 0 0 0 0 0 0 0 0 0  0  0  0  0 0 0  0  0  0  0 0
```

Pelas linhas iniciais do arquivo acima (cabeçalho), podemos notar que a imagem possui resolução 24x7 (24 colunas e 7 linhas) e o valor máximo da escala de cinza é 63, isto é, os valores de cinza variam entre 0 e 63 (64 tons de cinza).

Utilize as imagens disponibilizadas na página da disciplina para realizar o teste do programa. Você pode utilizar o programa GIMP (http://www.gimp.org/) ou OpenSeeIt (http://openseeit.sourceforge.net/) para visualizar as imagens.

Operador de detecção de bordas de Sobel

Vide: http://pt.wikipedia.org/wiki/Filtro_Sobel

O operador de Sobel realiza um cálculo para cada pixel da imagem de entrada, gerando assim um pixel da imagem de saída.

Para calcular o novo valor de um pixel da imagem, utilize a seguinte fórmula:

novo_z5 = ((z7 + 2*z8 + z9 - (z1 + 2*z2 + z3)) * (z7 + 2*z8 + z9 - (z1 + 2*z2 + z3)) +
(z3 + 2*z6 + z9 - (z1 + 2*z4 + z7)) * (z3 + 2*z6 + z9 - (z1 + 2*z4 + z7))
)**0.5

onde, considerando que o novo pixel a ser processado corresponde ao pixel central z5, os demais seguem a seguinte vizinhança:

z1	z2	z3
z4	z5	z6
z7	z8	z9

Isto é, se o pixel a ser calculado (z5) possuir coordenadas (i, j), os seus vizinhos serão os elementos das coordenadas a seguir:

z1: (i-1, j-1)	z2: (i-1, j)	z3: (i-1, j+1)
z4: (i, j-1)	z5: (i, j)	z6: (i, j+1)
z7: (i+1, j-1)	z8: (i+1, j)	z9: (i+1, j+1)

Portanto, dada a matriz com os pixels do arquivo de entrada, será criada uma nova matriz com os pixels calculados pelo operador Sobel. A fórmula acima será aplicada a cada posição da matriz de entrada considerando os seus elementos vizinhos e o valor resultante será armazenado na mesma posição da matriz de saída. Todos os cálculos são realizados com os valores da matriz de entrada e os resultados são armazenados na matriz de saída.

Se o resultado do cálculo de um pixel ultrapassar o valor máximo de tons de cinza (63 no exemplo acima), ele deverá ser limitado a este valor.

Se um pixel não tiver algum dos seus vizinhos, os valores destes vizinhos devem ser considerados como zero na fórmula. Isto pode ocorrer com os pixels da primeira e última linha, assim como da primeira e última coluna.