UNIVERSIDADE FEDERAL DE MINAS GERAIS
Departamento de Ciência da Computação

Disciplina	Turmas
Algoritmos e Estruturas de Dados II	

Professores

Data de entrega: 18/04/2016 até as 23:55

Execício de Programação em Linguagem C (TP0)

O objetivo deste trabalho é implementar um detector de objetos por meio de uma janela deslizante e a função de correlação cruzada entre duas imagens no formato PGM. Uma imagem PGM tem o formato abaixo, onde P2 é apenas o indicativo que o formato contém uma imagem em tons de cinza, c indica o número de colunas da imagens (largura da imagem), l indica o número de linhas da imagem (altura da imagem) e v indica o valor máximo que um pixel pode assumir (neste trabalho considera-se que v será no máximo 255, para poder ser armazenado em um *unsigned char*). As entradas p(i,j) indicam o valor do pixel na linha i e coluna j da imagem.

```
P2
c l v
p(0,0) p(0,1) ... p(0,c-1)
p(1,0) p(1,1) ... p(1,c-1)
p(2,0) p(2,1) ... p(2,c-1)
...
p(l-1,0) p(l-1,1) ... p(l-1,c-1)
```

Um exemplo de imagem é dado a seguir.

```
P2
3 2 255
200 128 24
255 211 35
```

Esta imagem contém três pixels de largura e dois pixels de altura, o que equivale a uma matriz com 2 linhas e 3 colunas, e o valor máximo de um pixel é de 255. Neste exemplo, p(1,2) = 35 e p(0,0) = 200.

Estrutura de dados

A estrutura de dados a seguir deve ser utilizada para armazenar a imagem após carregada do arquivo.

```
typedef struct {
   int c;                    // número de colunas na imagem
   int l;                    // número de linhas na imagem
   unsigned char maximo;     // valor máximo para cada pixel
   unsigned char **dados;    // variável para armazenar os pixels da imagem (matriz)
} PGM;
```

A estrutura de dados a seguir deve ser utilizada para armazenar um ponto (x,y).

```
typedef struct {
   int x;              //coluna
   int y;      //linha
} Ponto;
```

Correlação Cruzada

A correlação cruzada é uma medida de similaridade entre duas séries em função do deslocamento relativo entre ambas. Esta também é conhecida como produto interno deslizante. É comumente usada para encontrar em um sinal longo por uma característica de menor comprimento. Uma imagem digital é uma função bidimensional discreta *f(x,y)* que retorna a intensidade do pixel (entre 0 e 255) dadas as coordenadas x (coluna) e y (linha). Assim, a função de correlação cruzada para uma imagem discreta é definida como:

$$h[x,y] = f[x,y] \star g[x,y] = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} f[n_1,n_2] \cdot g[x+n_1, y+n_2] \tag{1}$$

Em processamento de imagens, a correlação cruzada pode ser utilizada como técnica para encontrar um objeto na imagem. Neste caso, tem-se uma imagem de entrada **g** e a imagem de um objeto **f** que queremos encontrar. O algoritmo da janela deslizante consiste em calcular a função **h** conforme ilustrado na figura abaixo:

1	5	0	8
0	7	10	5
6	2	4	7

Imagem **g**

10	5
4	7

Objeto **f**

h(0,0) = [1 5; 0 7] * [10 5; 4 7] = 1*10 + 5*5 + 0*4 +7*7 = 84

h(0,1) = [5 0; 7 10] * [10 5; 4 7] = 5*10 + 0*5 + 7*4 + 10*7 = 148

h(0,2) = [0 8; 10 5] * [10 5; 4 7] = 0*10 + 8*5 + 10*4 + 5*7 = 115

h(1,0) = [0 7; 6 2] * [10 5; 4 7] = 73

h(1,1) = [7 10 5; 2 4 7] * [10 5; 4 7] = 156

h(1,2) = [10 5; 4 7] * [10 5; 4 7] = 190

h(1,2) = 190 é o maior valor e é onde ocorre o match!!

Pela figura acima, pode-se observar que a janela deslizante percorre os pixels de **g** e, em cada pixel (x,y), calcula-se a correlação cruzada entre a imagem **f** e uma região de **g** iniciando no ponto (x,y) e com mesmo tamanho de **f**. Se **f** possuir tamanho $m \times n$ e **g** possuir tamanho $p \times q$. Então a função **h** é definida como:

$$h[x,y] = \sum_{j=0}^{m-1} \sum_{i=0}^{n-1} f[j,i] \cdot g[x+j, y+i], \tag{2}$$

onde $0 \leq x \leq m-p$ e $0 \leq y \leq n-q$. Os limites superiores para x e y previnem que haja estouro no acesso de memória, pois conforme na equação acima: $g[x+j, y+i]$ pode chegar no máximo até a posição correspondente à $g[m-1, n-1]$.

O que deve ser feito

1. Elaborar um programa capaz de ler duas imagens no formato PGM e imprimir no arquivo a posição x y onde a função de correlação cruzada retornou o maior valor. A linha de comando será a seguinte.

   `./exec imagem_cena.pgm imagem_objeto.pgm imagem_saida.txt`

   onde *exec* é o nome do executável, *imagem_ cena.pgm* contém o nome da imagem da cena de entrada (imagem no formato PGM), *imagem_ objeto.pgm* contém o nome da imagem do objeto de entrada (imagem no formato PGM), *imagem_ saida.txt* receberá a saída do algoritmo contendo apenas a posição x y encontrada pelo algoritmo.

2. O programa deve ter a função `PGM *LePGM(char* entrada)` que lê uma imagem no formato PGM do arquivo *entrada* e armazena seus dados em uma variável do tipo PGM.

3. O programa deve ter a função `int CorrelacaoCruzada(PGM *cena, PGM *obj, Ponto p)` que calcula a correlação cruzada entre a imagem da cena *cena* na posição p = (x,y) e a imagem do objeto *obj* e retorna este valor.

4. O programa deve ter a função `Ponto JanelaDeslizante(PGM *cena, PGM *obj)`, que aplica o algoritmo da janela deslizante calculando a correlação cruzada entre as regiões da cena e o objeto, retornando o ponto (x,y) onde houve maior valor.

5. Todas as funções implementadas devem possuir um cabeçalho conforme o exemplo a seguir:

```
/*-----------------------------------------------------------------
Protótipo: CorrelacaoCruzada(PGM *cena, PGM *obj, Ponto p)
Função: Aplicar a correlação ...
Entrada: Estrutura contendo ....
Saída: ...
-----------------------------------------------------------------*/
```

6. Seu programa não deverá ter nenhum *leak* de memória, ou seja, tudo que for alocado de forma dinâmica, deverá ser desalocado com a função "free()"antes do término da execução.

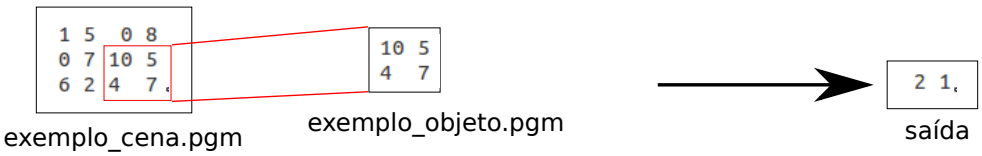
7. O programa deverá ser possível de ser compilado no Linux, portanto ele não deverá conter nenhuma biblioteca que seja específica do sistema operacional Windows.

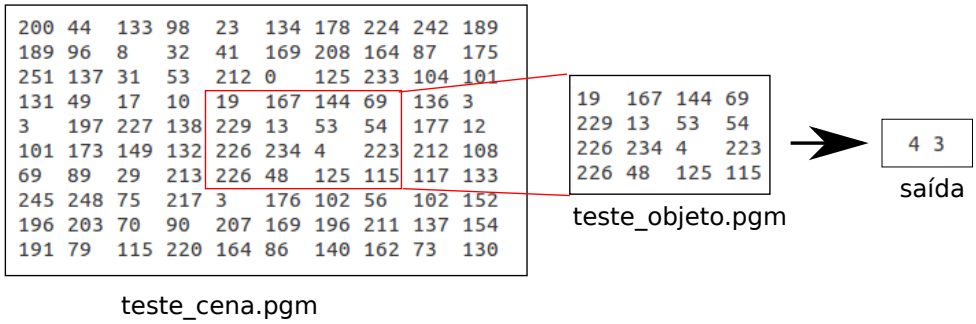
8. Você deverá implementar o trabalho na IDE Code::Blocks (*disponível em: http://www.codeblocks.org/*). Deve ser submetido ao moodle o diretório do projeto criado no Code::Blocks em um arquivo compactado contendo os arquivos ".h"(com as declarações das funções para ler e salvar imagens PGM) e ".c"(um com as implementações das funções descritas nesse documento e outro com a função *main*. Além disso, a submissão também deverá conter o relatório do trabalho em formato PDF.

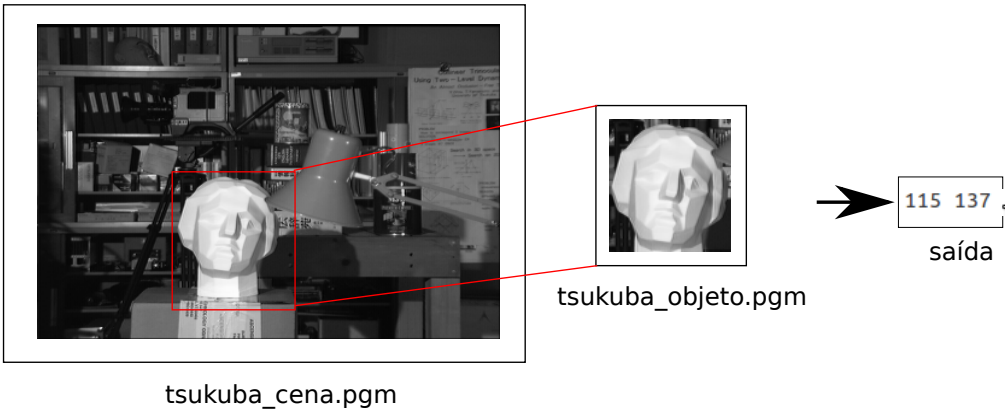
9. O trabalho deverá ser entregue via *moodle* até as 23:55 horas do dia 18/4/2016. **Trabalhos que forem entregues fora do prazo não serão aceitos em nenhuma hipótese.**

Exemplo para teste

Para avaliar o funcionamento do seu programa disponibilizamos as imagens: *exemplo_ cena.pgm* / *exemplo_ objeto.pgm*, *teste_ cena.pgm* / *teste_ objeto.pgm* e *tsukuba_ cena.pgm* / *tsukuba_ objeto.pgm*. Os resultados deverão ser:







Documentação

Escreva um documento explicando o seu código e avaliando o desempenho de sua implementação. Separe-o em cinco seções: introdução, implementação, resultados, conclusão e referências. Seja claro e objetivo.

- **Introdução**: Faça uma breve introdução o problema, definindo-o com as *suas* palavras.
- **Implementação**: Explique quais foram as estratégias adotadas para a leitura das imagens em formato PGM, realização da operação de convolução e a escrita da imagem PGM em disco. Descreva qual o papel de cada função, seus parâmetros de entrada e de saída.

- **Resultados**: Faça testes com seu programa utilizando diversas imagens. Apresente o resultado obitido e faça uma breve discussão sobre eles. (Serão disponibilizadas duas imagens para teste)
- **Conclusão**: Explique quais foram as dificuldades encontradas durante o desenvolvimento e as conclusões obtidas.
- **Referências**: Se você utilizou informações adicionais além das especificadas neste trabalho, cite as fontes.

Avaliação

- **4 pontos** pela implementação, onde serão avaliados, além do funcionamento adequado do programa: identação correta do código, comentários das funções, alocações de desalocações dinãmicas bem feitas e modularização.
- **5 pontos** pela documentação, onde serão avaliados a clareza das seções e a discussão dos resultados obtidos.
- ***Lembramos que será utilizado um software de detecção de cópias e qualquer tipo de plágio detectado resultará em nota 0 para ambos trabalhos!***

Pontos Extras

Todos os alunos da UFMG são bons programadores, mas sabemos que alguns destes alunos se destacam entre os demais. Para que estes alunos não fiquem entediados após terminarem este TP, vamos distribuir 1 (um) ponto extra em algumas tarefas. **Nota.** Você precisa ter feito todo o TP para poder receber os pontos extras!

(1 ponto): Se usarmos como imagem da cena o arquivo *exemplo_ cena.pgm* e pesquisarmos pela imagem do objeto descrito abaixo:

```
P2
2 2 255
1 5
0 7
```

Qual o resultado do algoritmo? Discuta brevemente sobre a falha ocorrida e proponha uma modificação na implementação criando uma nova função com escopo *int CorrelacaoModificada(PGM *cena, PGM *obj, Ponto p)* que retorna o resultado correto. **Obs:** a função *CorrelacaoModificada* poderá retornar um *float* caso necessário.

Para utilizar a nova função, deve-se criar um quarto parâmetro opcional na linha de comando que receberá:

- *0* - para usar a correlação cruzada
- *1* - correlação modificada.

Por exemplo:

```
./exec imagem_cena.pgm imagem_objeto.pgm imagem_saida.txt 1
```

Se não estiver definido nenhum valor para o quarto parâmetro, seu programa deverá usar a correlação cruzada como definido previamente no enunciado.