# Documentação Trabalho Prático 0 de AED's 2

**Aluno: Marcus Vinicius de Oliveira**

## 1. Introdução

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 defida como:

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

## 2. Implementação

O programa recebe através da linha de comando duas imagens no formato PGM e imprime a posição x y onde a função de correlação cruzada retornou maior valor.

Iniciamos a função main declarando as variáveis que serão utilizadas. Entre elas dois ponteiros de estrutura PGM, *imagem e *objeto. Estes recebem os dados das imagens cena e objeto através da função LePGM (char* entrada).

Após a leitura e armazenagem dos dados das imagens, a função JanelaDeslizante(PGM *cena, *objeto) é chamada, sendo as imagens supracitadas passadas como parâmetro. A função JanelaDeslizante(PGM *cena, *objeto), calcula todos as possíveis relações cruzadas entre a cena e o objeto. Além disso, ela armazena e retorna o ponto em que houve o maior valor de correlação cruzada.

O resultado da função é armazenado em uma variável Ponto chamada pontoMaiorCor. As estruturas PGM imagem e objeto já não são necessárias e por isso é feito sua liberação na memória através da função LiberaPGM(PGM *imagem).

Utilizando o terceiro argumento do argv[], um arquivo de saída é criado contendo a posição armazenada pela variável pontoMaiorCor. Em seguida o arquivo é fechado e o programa finalizado.

### 2.1 Funções

#### LePGM (char* entrada);

A função recebe como entrada uma imagem .PGM e retorna uma estrutura PGM contendo os dados da imagem.

Trivialmente a função declara as variáveis que serão utilizadas, em especial a variável PGM *imagem. É alocado então uma estrutura do tipo PGM, ao qual *imagem passa a apontar.

É lido e computado parte do arquivo recebido como parâmetro. Logo, com os dados necessários na memória, é alocado uma matriz em imagem.dados. Posteriormente é

retomada a leitura e armazenagem dos dados da imagem na estrutura PGM. Está estrutura é retornada pela função.

### Void LiberaPGM(PGM *imagem);

A função recebe como entrada uma estrutura PGM e a libera da memória com um uma serie de aplicações da função free.

### Int CorrelacaoCruzada(PGM *cena, PGM *obj, Ponto p);

Está é função mais complexa do programa. O estudo de sua natureza não é o foco de nosso trabalho. Nosso trabalho apenas a implementou em linguagem C a partir das explicações, definições e da equação (1) da especificação do TP0.

Dada duas matrizes n x m, a correlação cruzada entre elas é dada pela soma dos produtos dos elementos [i][j] das mesmas.

Sua definição não é muito complexa, mas sua implementação não é trivial. É preciso lembrar que uma das matrizes n x m (a matriz cena), na verdade é uma sub-matriz e que os valores de i e j são relativos ao da outra matriz (matriz objeto). Neste sentido é preciso utilizar de artifícios para encontrar essa relação.

A estratégia utilizada foi simples:

Naturalmente, são usados dois for's para percorrer a matri:

O primeiro for (i=p.x; i<(p.x+obj->l) ;i++) inicia-se com i =p.x devido ao valor do ponto inicial. Ele irá percorrer o numero de linhas que a matriz objeto possui (p.x+obj->l).

O Segundo for (j=p.y;j<(p.y+obj->c); j++) tem a mesma logica. Inicia-se com j=p.x devido ao valor do ponto inicial e irá percorrer o numero de colunas que a matriz objeto possui(p.y+obj->c).

Dentro do segundo for que ocorre a soma dos produtos. A estratégia foi argilosa: para matriz da cena, consideramos os valores brutos de i e j. Dessa forma a correlação ser calculada a partir do ponto solicitado. Para a matriz do objeto, i = i – p.x e j = j - p.y . Dessa maneira o inicio da correlação será sempre no início do objeto.

### Ponto JanelaDeslizante (PGM *cena, PGM *obj);

Esta função recebe as duas estruturas PGM como parâmetro e retorna o ponto onde a correlação cruzada é maior. Seu funcionamento é trivial. São dois for's que fazem com que a matriz objeto deslize sobre a matriz cena. Os for's são limitados pelas diferenças das linhas e colunas das matrizes.

Dentro do segundo for ocorre a chamada da função CorrelacaoCruzada para cada correlação possível para as duas matrizes. Ocorre também o armazenamento do ponto com maior correlação cruzada. Esse ponto é retornado ao final da chamada da função.

## 3. Resultado

O programa rodou sem problemas para os três casos testes propostos. O numero de testes foi pouco para uma analise mais detalhada. Caso houvessem mais casos, uma analise do tempo de execução poderia ser feita.

exemplo_cena  exemplo_objeto → saída: 2 1

teste_cena teste_objeto → saída: 4 3

tsukuva_cena tsukuba_objeto → saída: 115 137

## 4. Conclusão

Durante a implementação uma dificuldade inicial com parâmetros da argv[] bem como uma dificuldade com ponteiros foi sentida, entretanto com auxilio da comunidade stackoverflow, as dificuldades foram vencidas e o trabalho continuou sem muita dificuldade.

## 5. Referências


http://pt.stackoverflow.com/questions/4858/função-retorna-ponteiro-para-lixo-e-free-trava-o-terminal

http://stackoverflow.com/questions/15860628/reading-from-a-file-into-the-argv-array

http://pt.stackoverflow.com/questions/4858/função-retorna-ponteiro-para-lixo-e-free-trava-o-terminal