



**Escola Politécnica da Universidade de São Paulo**

PSI3471 - Fundamentos de Sistemas Eletrônicos Inteligentes

## Relatório EP

**Integrantes:**

Gustavo Oliveira Barranova

Gustavo Misawa Hama

**No USP:**

10336661

10773461

**Docente(s):**

Hae Yong Kim

## Breve enunciado do problema

O exercício-programa proposto aos alunos da disciplina PSI3471 (Fundamentos de Sistemas Eletrônicos Inteligentes) possui como objetivo técnico identificar o modelo do canhoto, sendo 3 opções e alinhar a imagem do canhoto com a imagem do canhoto padrão resultante. São 21 imagens ao todo, com diferentes modelos e tamanhos.

## Ambiente de desenvolvimento utilizado

O código foi desenvolvido na linguagem C++, e no editor VSCode (Visual Studio Code). O código utiliza a biblioteca cekeikon. Essa biblioteca foi disponibilizada no próprio site da disciplina e é utilizada amplamente para a leitura e manipulação de imagens.

O programa pode ser compilado a partir do próprio compilador da biblioteca cekeikon, e pode ser executado no “prompt de comando”. A compilação do programa segue a seguinte sintaxe:

```
compila ep -c -v3
```

## Técnicas utilizadas para resolver o problema

A abordagem adotada para a resolução do problema foi praticamente a mesma sugerida no enunciado do exercício-programa, e será descrita a seguir detalhando as etapas seguidas.

### 1. Manipulação da imagem original

A imagem lida pelo programa é originalmente colorida. Portanto, para manipular a imagem trataremos ela com tons de cinza, lendo como Mat\_<FLT>.

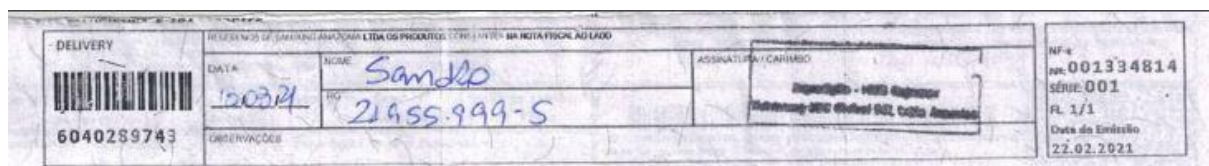


Figura 1: Exemplo de imagem original disponibilizada pelo enunciado. Possui nome original “m2\_001.jpg”.

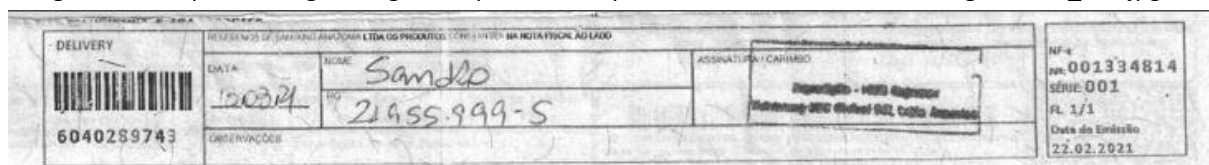


Figura 2: Imagem modificada que apresenta as cores em tons de cinza

Como os canhotos modelos possuem 750 colunas, redimensiona-se a imagem para ficar com 750 colunas e as linhas para manter o aspect ratio.

## 2. Customização e redimensionamento dos modelos do canhoto

Para utilizar o match template e percorrer corretamente a imagem a ser analisada e as imagens modelos são emolduradas de tal forma que as imagens modelos possuem tamanho menor que a analisada, 310x1150 e 210x1050, respectivamente. Para tal, utilizamos a função *fazmoldura* mostrada abaixo. Preenchemos a moldura da imagem a ser analisada com a média de cinzas e a dos canhotos modelos com 128.0/255.0 (don't care).

```
Mat_<FLT> fazmoldura(Mat_<FLT> x, int i, int j, float k){
    Mat_<FLT> y(i, j);
    int x1 = (i-x.rows)/2; //50
    int xc = (j-x.cols)/2; //150
    //printf("x1: %i, xc: %i, x.rows: %i, x.cols: %i\n", x1, xc, x.rows, x.cols);
    for (int l = 0; l < y.rows; l++){
        for (int c = 0; c < y.cols; c++){
            if((l >= x1 && c > xc) && (l >= x1 && c < (xc+x.cols)) && (l < (x1+x.rows) && c > xc)) y(l,c) = x(l-x1,c-xc);
            else y(l, c) = k;
        }
    }
    return y;
}
```

Figura 3: Implementação da função *fazmoldura*

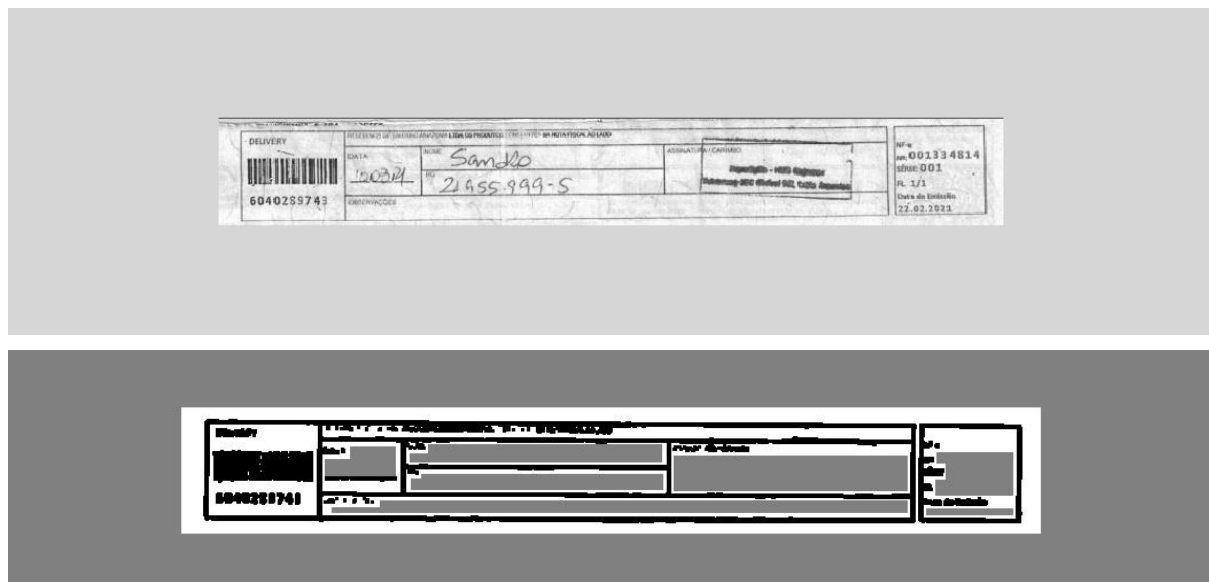


Figura 4 e 5: Imagens dos canhotos emoldurados

Obtido o modelo dos canhotos emoldurados, é realizada uma rotina no código no qual os modelos são redimensionados em escala de progressão geométrica, rotação e deslocamento.

Para a progressão multiplica-se de 0.88 a 1.09 de 0.1 a 0.1, para a rotação rotaciona-se de -3 a +3 graus de 0.375 cada passo. E para o deslocamento medimos a distância entre o ponto da matriz de maior valor de correlação e o centro da imagem.

## 3. Construção do loop de correlação por Template Matching

Utilizando-se dos modelos redimensionados em cada iteração do loop, o próximo passo é gerar as matrizes de correlação referentes a cada modelo analisado, guardando o valor da maior correlação em uma variável feita para cada modelo analisado. Para isso,

primeiramente, foi gerada uma matriz que utiliza a função *somaAbsDois(dcReject(<matriz\_de\_entrada>)*, disponível na biblioteca *cekeikon*. Isso é necessário para realizar o casamento de modelo CC, o qual foi utilizado para a resolução do problema.

Fazemos isso para cada um dos modelos e no final comparamos para ver qual o modelo que possui a melhor correlação. Para o deslocamento pega-se o ponto onde há a maior correlação e subtrai-se do ponto onde está o meio da imagem, assim obtendo quanto está deslocado.

No final imprime-se o canhoto padrão resultante em cima da imagem de entrada.

```
Mat_<FLT> qfinal2 = fazmoldura(qfinal, 310, 1150, 128.0/255.0);
for(int i = 0; i < qfinal2.rows; i++){
    for(int j = 0; j < qfinal2.cols; j++){
        if(qfinal2(i, j) == 0){
            fotofinal(i,j)[2] = 255;
            fotofinal(i,j)[1] = max(fotofinal(i,j)[1] - 100,0);
            fotofinal(i,j)[0] = max(fotofinal(i,j)[0] - 100,0);
        }
        if(qfinal2(i, j) > 0.4 && qfinal2(i, j) < 0.6){
            fotofinal(i,j)[0] = 255;
            fotofinal(i,j)[1] = max(fotofinal(i,j)[1] - 100, 0);
            fotofinal(i,j)[2] = max(fotofinal(i,j)[2] -100, 0);
        }
    }
}
```

Figura 6: implementação da sobreposição

## Operação

O programa, após ser compilado conforme as instruções dadas anteriormente, pode ser executado no próprio “prompt de comando”, e não necessita de qualquer outra configuração adicional. Ele deve seguir a seguinte sintaxe:

ep <entrada> <saida>

Sendo assim, o programa exige que o usuário identifique o arquivo de entrada desejado, e também o nome do arquivo de saída que será gerado, seguido do seu formato. Caso entre com entradas inválidas o programa exibe mensagens como proposto no enunciado.

## Resultados obtidos

Concluída a implementação da resolução do problema, foi possível observar que o programa consegue identificar corretamente os 21 canhotos disponibilizados como exemplo. Entretanto, na sobreposição das imagens *m4\_002* e *m4\_009* não sobrepõe corretamente dado que a imagem dos canhotos estão muito ruins.



Figura 7: m2\_002.jpg sobreposta pelo canhoto 2

```
PS C:\Users\The Doctor\Documents\Poli\PSI3471\hae\gustavo_hama_gustavo_barranova> .\ep.exe m2_002.jpg ep_m2_002.jpg
Maior correlacao entre m2_002.jpg e m2.pgm: 0.171821
Maior correlacao entre m2_002.jpg e m3.pgm: 0.091886
Maior correlacao entre m2_002.jpg e m4.pgm: 0.070689
melhorModelo=2 corr= 0.171821 graus= 0.750000 fator= 1.005714 desloc(x,y)= [-2, -5]
28.712 s
```

Figura 8: Resultados no terminal



Figura 9: m3\_003.jpg sobreposta pelo canhoto 3

```
PS C:\Users\The Doctor\Documents\Poli\PSI3471\hae\gustavo_hama_gustavo_barranova> .\ep.exe m3_003.jpg ep_m3_003.jpg
Maior correlacao entre m3_003.jpg e m2.pgm: 0.083565
Maior correlacao entre m3_003.jpg e m3.pgm: 0.169966
Maior correlacao entre m3_003.jpg e m4.pgm: 0.103104
melhorModelo=3 corr= 0.169966 graus= -0.375000 fator= 1.037143 desloc(x,y)= [-1, -1]
28.661 s
```

Figura 10: Resultados no terminal

## Referências

O material consultado para a elaboração do trabalho se restringiu às apostilas da disciplina e ao enunciado do exercício programa, disponíveis em:

<http://www.lps.usp.br/hae/psi3471/index.html>.