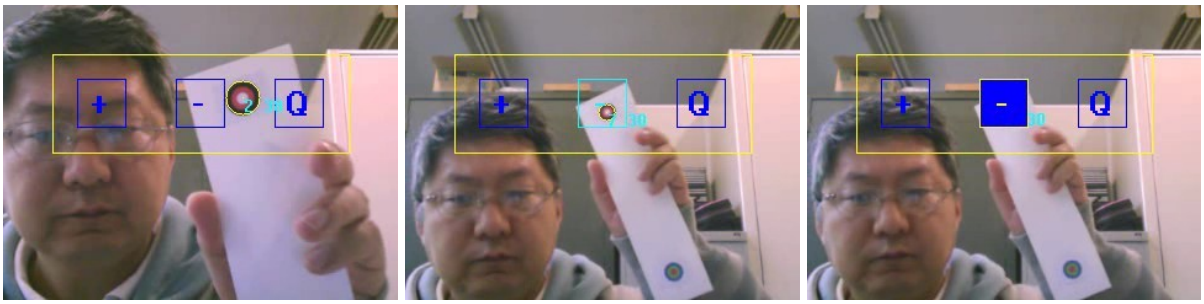
PSI-5796: Processamento e Análise de Imagens e Vídeos
Segundo período de 2015 **1º exercício-programa** **Prof. Hae**
Data de entrega: 02/08/2015 (domingo) até 24:00 horas
Entrega adiada para 09/08/2015 até 24:00 horas.

Obs. 1: Cada dia de atraso acarreta uma perda de 1 ponto no exercício.
Obs. 2: Este EP deve ser resolvido individualmente. EPs iguais receberão nota zero.

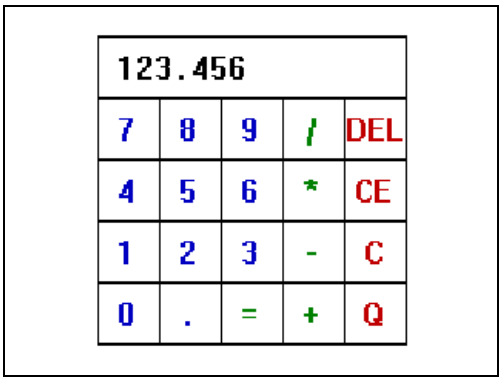
O vídeo abaixo mostra um "mouse virtual" utilizando um padrão circular e uma webcam (se não conseguir visualizar o vídeo, é necessário instalar FFDSHOW):

- http://www.lps.usp.br/~hae/psi5796/ep1-2015/cmatch.avi

O uso de padrão circular facilita a solução, pois é invariante por rotação. Se o padrão ficar parado num botão durante um certo tempo, considera-se que o usuário deu "click" nesse botão. O padrão pode estar localizado (aproximadamente) de 20 a 100 cm da webcam.



Faça um programa C/C++ que utiliza uma técnica semelhante para operar uma calculadora através da webcam. Os botões da calculadora podem ser, por exemplo:



Apertando botão 'Q', sai do programa.

Nota 1: Pode utilizar um padrão e/ou layout da calculadora diferente.

Nota 2: Você deve inverter esquerda-direita do quadro capturado. Caso contrário, o mouse moverá para a direção errada.

Nota 3: No vídeo-exemplo, estou procurando pelo padrão somente dentro do retângulo amarelo. Isto acelera o processamento.

Nota 4: Em Cekeikon, as seguintes funções podem ser úteis:

- `void reta(Mat_<COR>& b, int l1, int c1, int l2, int c2, COR cor=COR(0,0,0), int largura=1);`
- `void retang(Mat_<COR>& a, int l1, int c1, int l2, int c2, COR cor=COR(0,0,0), int largura=1);`
- `void putTxt(Mat_<COR>& ap, int la, int ca, string st, COR fore=COR(0,0,0), int ampl=1, bool transp=true, COR back=COR(255,255,255));`

Essas funções respectivamente traça uma reta, desenha um retângulo, e imprime um texto. As funções semelhantes do OpenCV são:

- `void line(Mat& img, Point pt1, Point pt2, const Scalar& color, int thickness=1, int lineType=8, int shift=0);`
- `void rectangle(Mat& img, Point pt1, Point pt2, const Scalar& color, int thickness=1, int lineType=8, int shift=0);`
- `void putText( Mat& img, const string& text, Point org, int fontFace, double fontScale, Scalar color, int thickness=1, int lineType=8, bool bottomLeftOrigin=false );`

Nota 5: O seguinte programa foi usado para imprimir o padrão "circulo.png":

```
#include <cekeikon.h>
int main()
{ int n=401;
  ImgXyb<COR> a(n,n,COR(255,255,255));
  a.centro(); //centro no centro da imagem
  for (int x=a.minx; x<=a.maxx; x++)
    for (int y=a.miny; y<=a.maxy; y++) {
      double r=sqrt(double(x*x+y*y));
      if (r<=50) a(x,y)=COR(255,255,255);
      else if (r<=100) a(x,y)=COR(0,0,255);
      else if (r<=150) a(x,y)=COR(0,0,0);
    }
  imp(a,"circulo.png");
}
```

Obs. 1: Pode usar (se quiser) a biblioteca Cekeikon/OpenCV. Um exemplo de captura de webcam está em c:\cekeikon3\samples\crt\webcam.cpp. Um outro exemplo está em c:\cekeikon3\samples\win32\showcam.cpp.

Obs. 2: Entregue o programa-fonte (ep1.cpp) e um documento PDF (coment.pdf) ou DOC (coment.doc) com os comentários descrevendo em português o funcionamento do programa. O envio dos comentários é obrigatório.

(a) Se você fez o programa no ambiente usado na classe (cekeikon/opencv): não é necessário enviar o programa executável.

(b) Se você usou um ambiente diferente: É necessário descrever como gerar o executável a partir do código fonte. É também necessário enviar executável. Mude a extensão .EXE para .EEE pois há servidores que não aceitam enviar/receber ".EXE". Um programa .EXE pode necessitar de vários arquivos .DLL para funcionar - envie todos os .DLLs necessários.

Obs. 3: Se você usou algum padrão diferente do circulo.png acima, pode enviar o padrão impresso físico para o monitor testar.

Obs. 4: Compacte todos os arquivos como **`SeuNome_Sobrenome.ZIP`** e envie um email colocando como assunto **"PSI5796 EP1"** para os endereços abaixo:

- **`hae@lps.usp.br`**
- **`gerson.faria@gmail.com`**

Para evitar confusão, envie um único email. Se você enviar dois ou mais emails, considerarei somente o último email enviado, descartando os anteriores.

O monitor da disciplina, Gerson Faria, escreveu o seguinte documento sobre os itens que deve conter o "coment.doc". Concordo plenamente com o que ele escreveu. Sugiro a todos que sigam as instruções dadas abaixo.

Tópicos mínimos exigidos no relatório (ou comentário) dos Exercícios Programas

1. Descrição do problema / objetivos

Descreva claramente o enunciado do problema a ser resolvido. Isso é importante para você se assegurar de que está resolvendo o problema pedido.

2. Técnica(s) utilizada(s) para resolver o problema

Descreva de forma clara quais algoritmos e técnicas foram necessários para resolver o problema. Utilize o nome adequado, se existente (e.g. filtragem Gaussiana, classificador SVM, algoritmo Sift etc.). Use elementos gráficos como imagens e diagramas se necessário. Não copie e cole código fonte, a não ser que o mesmo seja de fato relevante, mas se o fizer, comente-o. No relatório, o comentário é mais importante do que o código. Prefira o uso de pseudocódigo.

3. Ambiente de desenvolvimento utilizado

Em qual plataforma a solução foi desenvolvida? Em qual plataforma a solução será utilizada? Como o usuário pode compilar o programa? Quais bibliotecas foram utilizadas?

4. Operação

Como o usuário deve executar o programa? Quais os argumentos para execução? Há parâmetros necessários a serem configurados? Quais arquivos de entrada são necessários? Quais arquivos de saída são gerados?

5. Resultados Obtidos

Descreva os resultados obtidos. O problema foi resolvido de forma satisfatória / robusta? Quais as limitações encontradas? Quais as sugestões de melhorias?

6. Referências

Descreva o material externo utilizado, como livros consultados, websites visitados etc.