PSI-3471: Fundamentos de Sistemas Eletrônicos Inteligentes
Primeiro semestre de 2017 1º exercício-programa Prof. Hae
Data de entrega: 29/04/2017 (sábado) 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.

São dadas 16 imagens qr??.jpg, onde em cada imagem aparece um único QR-code. Os QR-codes aparecem aproximadamente de pé, com pequenas distorções em perspectiva. Essas imagens estão disponíveis em:

http://www.lps.usp.br/hae/psi3471/ep1-2017/qr.zip

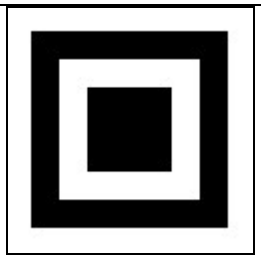
qr00.jpg


qr01.jpg

Faça um programa C/C++ ep1.cpp que lê uma das imagens qr??.jpg e gera uma imagem lo??.png que localiza as três marcas quadradas que permitem localizar o qr-code. Por exemplo, o comando abaixo deve ler qr00.jpg e gerar lo00.png:

```
c:\diretorio>ep1 qr00.jpg lo00.png
```

lo00.png

Pode usar, se quiser, a imagem “padrão.png”. http://www.lps.usp.br/hae/ psi3471/ep1-2017/padrão.png	 padrao.png

Opcional (o exercício vale 10,0 mesmo sem resolver este item):

Além disso, são dadas 5 imagens op??.jpg, onde em cada imagem aparece um QR-code e partes quebradas de outros QR-codes. Essas imagens estão disponíveis em:

http://www.lps.usp.br/hae/ psi3471/ep1-2017/op.zip

 op00.jpg	 op04.jpg

Opcionalmente, faça um programa C/C++ op1.cpp que lê uma das imagens op??.jpg e gera uma imagem op??.png que localiza as três marcas quadradas que aparecem no QR-code inteiro, desprezando as marcas quadradas dos QR-codes parciais. Por exemplo, o comando abaixo deve ler op00.jpg e gerar op00.png:

```
c:\diretorio>ep1-op op00.jpg op00.png
```

op00.png

Nota 1: 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 2: Para achar um padrão com tolerância a mudança de escala, deve procurá-lo depois de redimensioná-lo para várias escalas. Para redimensionar uma imagem, use a função do OpenCV resize:

```
void resize(InputArray src, OutputArray dst, Size dsize, double fx=0, double fy=0, int interpolation=INTER_LINEAR)
```

Veja o manual de referência de OpenCV para mais detalhes desta função. Veja a apostila sobre reamostragem (reamost.doc ou reamost.pdf) para mais detalhes.

Obs. 1: Pode usar (se quiser) a biblioteca Cekeikon/OpenCV.

Obs. 2: Entregue o programa-fonte (ep1.cpp) e um documento PDF (relatorio.pdf) ou DOC (relatorio.doc) com os comentários descrevendo o funcionamento do programa. O envio do relatório é obrigatório (veja o anexo).

(a) Se você fez o programa no ambiente usado na classe (cekeikon/opencv): não é necessário enviar o programa executável. Se você fez o programa usando só OpenCV em Linux ou Mac, também não é necessário enviar o programa executável (desde que você não utilize nenhuma função exclusiva desses sistemas).

(b) Em princípio, não é permitido resolver usando outros ambientes. Converse com o professor para verificar o que é permitido e o que não é.

Obs. 3: Compacte todos os arquivos como **`SeuNome_Sobrenome.ZIP`** e envie via edisciplinas: https://edisciplinas.usp.br/course/view.php?id=36751

Obs. 4: Qualquer dúvida, entre em contato com o professor:

- **`hae@lps.usp.br`**

Anexo: Tópicos exigidos no relatório dos Exercícios Programas

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.

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 existir (por exemplo, filtragem Gaussiana, classificador SVM, algoritmo SIFT, etc.). Use elementos gráficos como imagens intermediárias e diagramas se necessário. Não copie e cole código fonte, a não ser que o mesmo seja de fato relevante. Se o fizer, comente-o. No relatório, o comentário é mais importante do que o código. Prefira o uso de pseudocódigo.

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?

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?

Resultados Obtidos

Descreva os resultados obtidos. Qual é o tempo de processamento típico? O problema foi resolvido de forma satisfatória / robusta? Quais as limitações encontradas? Quais as sugestões de melhorias?

Referências

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