

PSI-5796: Processamento e Análise de Imagens e Vídeos
Segundo período de 2017 **1º exercício-programa**
Data de entrega: 23/07/2017 (domingo) até 24:00 horas

Prof. Hae

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 banco de dados abaixo contém imagens de placas de trânsito, algumas delas com sujeira:

http://inf-server.inf.uth.gr/~gpotamianos/traffic_sign_database.html

Desse banco de dados, peguei 44 imagens com a placa "proibido virar" (à esquerda ou à direita) e deixei em:

http://www.lps.usp.br/hae/psi5796/ep1-2017/proibido_virar.zip



4.jpg



8.jpg

Faça um programa C/C++ ep1.cpp que lê uma dessas 44 imagens e detecta (localiza) a placa "proibido virar", reconhecendo se é proibido virar à esquerda ou à direita. Forma de chamar o programa deve ser:

```
diretorio>ep1 1.jpg 1-sai.jpg
```



4-sai.jpg



8-sai.jpg

Você pode escolher até 4 (das 44 imagens) para servirem de "amostras de treinamento" ou "modelos" a serem procurados. Neste caso, deixe explícito nos comentários quais foram as imagens escolhidas para servirem de "modelo".

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

Obs. 2: Entregue o programa-fonte (ep1.cpp) e um documento PDF (relatorio.pdf) 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) Se você quiser usar alguma biblioteca diferente, converse antes com o professor.

Obs. 3: Compacte todos os arquivos como SeuNome_Sobrenome.ZIP e envie um email para:

- hae@lps.usp.br

Se você enviar mais de um email, considerarei somente o último email enviado, apagando os anteriores.

Anexo: Relatórios dos exercícios programas

A primeira regra para uma comunicação clara é escrever para a sua audiência (<http://www.plainlanguage.gov/howto/guidelines/FederalPLGuidelines/think.cfm>). No caso do relatório, a sua audiência será o professor ou o monitor que irá corrigir o seu exercício. Assim, você não precisa escrever informações que são óbvias para professor/monitor. Por outro lado, precisa escrever aqueles dados que ajudarão professor/monitor a avaliar corretamente a sua solução. Seja breve. Passe as informações necessárias usando menos palavras/páginas possível.

Identificação

Seu nome, número USP, nome da disciplina, etc.

Breve enunciado do problema

Apesar do enunciado do problema ser conhecido ao professor/monitor, descreva brevemente o problema que está resolvendo. Isto tornará o documento compreensível para alguma pessoa que não tem o enunciado do EP à mão.

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

Descreva quais técnicas você usou para resolver o problema. Se você mesmo inventou a técnica, descreva em português a sua ideia, deixando claro que a ideia foi sua. Se você utilizar oração com sujeito indeterminado ou na voz passiva, o leitor pode não entender quem foi o responsável (Ex: "Criou-se um novo algoritmo" - Quem criou? Você? Ou algum autor da literatura científica? <http://www.plainlanguage.gov/howto/guidelines/FederalPLGuidelines/writeActive.cfm>). Se você utilizou alguma técnica já conhecida, utilize o nome próprio da técnica (por exemplo, filtragem Gaussiana, algoritmo SIFT, etc.) juntamente com alguma referência bibliográfica onde a técnica está descrita. Use elementos gráficos como imagens intermediárias e diagramas, pois ajudam muito a compreensão. Não copie e cole código-fonte, a não ser que seja relevante. Use preferencialmente o pseudo-código.

Ambiente de desenvolvimento utilizado

Em qual plataforma você desenvolveu o programa? Como o professor/monitor pode compilar o programa? Você utilizou que bibliotecas?

Operação

Como o professor/monitor pode executar o programa? Que argumentos são necessários para a execução do programa? Há parâmetros que devem ser 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?

Referências

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