

UNIVERSIDADE FEDERAL DE MINAS GERAIS — UFMG
INSTITUTO DE CIÊNCIAS EXATAS — ICEx
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO — DCC

Disciplina: DCC884/DCC043 — Visão Computacional
Professor: Mario Fernando Montenegro Campos (mario@dcc.ufmg.br)
Monitor: Erickson Rangel do Nascimento (erickson@dcc.ufmg.br)
Período: 1º semestre / 2010
Aulas: Terças e Quintas, das 9:25 às 11:05, na sala 2029 do ICEx
Página: <http://www.verlab.dcc.ufmg.br/cursos/verlab/visao/2010-1/index>

2ª Lista de Exercícios

Data da entrega: 11 de maio de 2010

A lista é individual e as datas de entrega são fixas, devendo o trabalho ser entregue no início da aula. O valor desta lista é 5 pontos. Para cada dia de atraso na entrega será descontado 0,5 ponto do valor da lista.

Não há formato específico para apresentar as respostas e resultados. Porém, clareza na redação é um requisito essencial. Embora o resultado possa ser manuscrito, lembre-se de que “o que não se pode ler não se pode avaliar”. Assim, sugere-se que o trabalho seja digitado, podendo o desenvolvimento matemático ser manuscrito.

O Matlab ou Scilab são recomendados para o desenvolvimento das aplicações, pois permitem o desenvolvimento rápido de aplicativos e já incorporam várias rotinas matemáticas e de manipulação de imagens. O Scilab é gratuito e pode ser baixado da página <http://www.scilab.org/>. Você também pode usar outras linguagens de programação, mas nesse caso lembre-se de incluir as instruções de compilação dos programas e eventuais arquivos auxiliares, como “makefiles” ou arquivos de projeto.

Em todos os casos, não se exige nenhuma interface gráfica: os programas podem ler os dados de entrada por argumentos de linha de comando ou por digitação pelo usuário, e podem gravar a saída em arquivos no disco.

Para arquivos gráficos, recomenda-se o uso dos formatos PNG ou JPEG. Nesse sentido o Matlab e o Scilab ajudam, pois ambos disponibilizam rotinas de leitura e gravação para esses formatos. Para os entusiastas de C e C++, a biblioteca FreeImage (<http://freeimage.sourceforge.net/>) é uma alternativa gratuita e multi-plataforma. Para a visualização dos arquivos, os aplicativos IrfanView (Windows) e xv (Linux) são boas ferramentas e também são gratuitos.

Em qualquer situação, os programas devem ser entregues com documentação de uso ou devem ser auto-explicativos.

1 Exercícios Teóricos

1. Descreva o que é ruído em uma imagem.
2. Faça uma pesquisa sobre as fontes de ruídos no processo de formação das imagens. Liste e explique cada uma das fontes encontradas.

3. O que é uma filtragem linear e uma filtragem não linear? Por que a filtro da mediana é um exemplo de filtro não linear?
4. Procure saber e explique como é o processo de criação de uma máscara para um filtro passa-baixas utilizando o método dos mínimos quadrados no domínio espacial dada uma especificação no domínio da frequência.
5. Descreva um algoritmo para detectar as bordas de uma imagem e sua localização. Explique o que ocorre com a detecção de bordas e sua localização em imagens com muito ruído?

2 Exercícios Práticos

1. Usando o método de projeto de máscaras que você descreveu na seção anterior, projete uma máscara 5x5 para um filtro passa-baixas de Butterworth. Escolha uma frequência de corte adequada e use $n = 1$. Escolha uma imagem e a filtre com a máscara obtida. Faça também uma filtragem usando o filtro de Butterworth do tamanho da imagem no domínio da frequência. Discuta sobre a diferença entre o resultado pelas das duas filtrações.
2. Na OpenCV estão presentes dois extratores de características. Um baseado no trabalho de Shi e Tomasi (cvGoodFeaturesToTrack) e outro no algoritmo SURF (cvExtractSURF). Explique o princípio de cada um dos métodos explicitando suas vantagens e desvantagens. Implemente um rastreador simples utilizando como extrator de características o SURF (há um exemplo do uso das funções de extração e comparação de características usando SURF no diretório da OpenCV: samples/c/find_obj.c)

3 Exercícios de Pesquisa

1. Pesquise sobre os algoritmos *Seeded Up Robust Features* (SURF) e *Scale-Invariant Feature Transform* (SIFT). Liste as principais características de cada um dos algoritmos. Em quais situações o SIFT é mais adequado?