



## PROJETO I

### Datas propostas:

- Entrega dos arquivos 26/02/2013 (via moodle até as 23.50 hrs)

O objetivo do Projeto a é fixar os conceitos de morfologia matemática, segmentação e estimação de movimento. O projeto deve ser feito individualmente, usando MATLAB ou OpenCV como ferramentas de desenvolvimento.

### Espera-se do projeto:

A clara identificação do problema a ser resolvido. Descrição completa da solução proposta e resultados. Identificação objetiva e avaliação dos resultados.

### O que deve ser entregue?

- Relatório feito pelos alunos, em 3 páginas como mínimo e 5 como máximo. Detalhes sobre o relatório, ler o arquivo "LER ANTES DE FAZER OS RELATÓRIOS" no Moodle.
- Código da solução, comentado e com descrição de uso e extensão.

### Questão 1

Faça um script em MATLAB ou em C/C++ que realize os seguintes passos:

1. Ler a imagem "cookies.tif"
2. Binarizar a imagem de tal forma que sejam identificados as duas "cookies" (escolher o limiar apropriado, de forma de diferenciar as cookies do fundo).
3. Eliminar por completo a "cooky" mordida, deixando pelo menos parte da cooky completa, na imagem binarizada. Indicar as operações morfológica utilizadas e elementos estruturantes no relatório. No relatório mostrar também a imagem resultante.
4. Recuperar a forma inicial da "cooky" completa na imagem resultante do passo anterior. Indicar as operações morfológica utilizadas e elementos estruturantes no relatório. No relatório mostrar também a imagem resultante.
5. A partir da imagem original e utilizando a imagem resultante do passo anterior como máscara, obter uma imagem final em níveis de cinza com somente a "cooky" completa.

## Questão 2

A estimação de movimento é uma das principais fontes de informação temporal de um vídeo. Dentre as diversas técnicas de estimação, o *Block Matching* oferece uma solução de relativa simplicidade. A sua principal característica é o uso de uma vizinhança quadrada (bloco) a partir da qual define-se uma região de busca na imagem adjacente e procura-se o deslocamento que minimiza uma medida de erro entre o bloco atual e o bloco deslocado na imagem adjacente.

Note, porém, que o uso de blocos está baseado na hipótese de que todos os pixels dentro dessa área respeitam o mesmo modelo de movimento translacional. Essa hipótese poderá ser violada quando o bloco contém uma fronteira de movimento como ilustrado na Figura 1.



Fig. 1 (a) Exemplo de imagem original e (b) ilustração de partição com bloco em fronteira de movimento.

Se assumirmos que as regiões de movimento homogêneo são compostas de sub-regiões de intensidade (ou cor) homogênea, podemos usar o resultado de uma segmentação baseada em intensidade (ou cor) para garantir que todos os pixels usados para comparação via *Block Matching* pertençam à mesma região de intensidade homogênea e, logo, de movimento homogêneo. A medida de erro usada para comparar blocos entre imagens adjacentes deverá alocar mais peso aos pixels que pertencem à mesma região do pixel central, como ilustra a vizinhança da Figura 2. Dessa forma, esperamos melhorar a precisão da estimativa de movimento em regiões de fronteira de movimento.

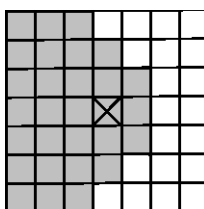


Fig. 2 Pixels do bloco (em cinza) que pertencem à mesma região do pixel central “x”.

1. Implemente um algoritmo de Block Matching capaz de o fluxo óptico, ou seja, vetores de movimento independentes para **todos os pixels da imagem**. Defina um tamanho de vizinhança  $N \times N$  ao redor do pixel, uma área para busca exaustiva  $S \times S$  na imagem adjacente, use como métrica para comparação a *Sum of Absolute Differences (SAD)*. Mostre no relatório o fluxo óptico, a imagem compensada, e indique a diferença entre PSNR da imagem compensada e a real. DICA: Fazer padding para tratar as bordas. Utilize os dois primeiros quadros do vídeo foreman.yuv
2. Escolha uma técnica de segmentação baseada em intensidade ou cor (e.g., a função *watershed* do MATLAB) e segmente a imagem atual. Note que as regiões segmentadas não devem ser muito menores que o tamanho do bloco usado no item 1 ( $N \times N$ ).
3. É possível afirmar que os pixels dentro dos objetos segmentados possuem vetores de movimento similares? Ou seja pixels que dentro do objeto possuem uma direção similar? Justifique.

**Imagens de teste:** Faça o download do arquivo de vídeo “foreman\_cif.yuv” do Moodle e extraia os primeiros dois quadros (#0 e #1) para imagens teste. Vide também arquivo MATLAB para leitura de yuv (`[y,u,v] = yuvRead('foreman.yuv',352,288,2);`).

**Dicas:** A função *quiver* do MATLAB é útil para plotar vetores.