Simulação e Modelação

Trabalho2

Este trabalho pretende analisar e modelar um fenómeno físico simples: a queda de uma bola no solo. A primeira parte consiste na análise do filme e extração de informação. Na segunda parte pretende-se modelar o fenómeno e determinar a aceleração da gravidade usando dois métodos diferentes.

PARTE 1

Vamos registar informação sobre o movimento de um corpo, a partir de um filme. Para melhor compreender como se manipulam imagens e vídeos, veja o Anexo 1 fornecido mais em baixo.

Passo 1: Leia o filme bola.mp4.

```
mov = VideoReader('bola.mp4');
video=read(mov);
s = size(video);
nFrames = s(4);
framerate = mov.FrameRate;
```

Passo 2: Para o primeiro fotograma (`video(:,:,:,1)`) isole o canal vermelho (ver Anexo 1). Aplique-lhe um filtro de limiar de forma a obter uma matriz binária, com o valor 1 no caso da cor corresponder à da bola, e 0 se corresponder à restante imagem. Chame a essa matriz Imagem.
Observe o resultado mostrando a imagem usando `imagesc(Imagem)`

Passo 3: Encontre os índices de coordenadas para os quais Imagem é igual a 1 usando a função find (`[xp,yp]=find(Imagem==1)`). Com a função mean, determine as coordenadas do centro da bola. Marque na imagem anterior o centro da bola calculado.

Passo 4: Com um ciclo "for", repita os passos 2 e 3 para todos os fotogramas numa animação. Guarde os valores da posição vertical da bola num vetor y. Nota: Como o vídeo tem uma framerate maior que a do seu monitor, utilize a instrução drawnow se quiser visualizar o vídeo com a representação do centro da bola determinado.

Passo 5: Usando o comprimento máximo de pixéis a 1, numa linha adequadamente escolhida, e sabendo que a bola tem 5 cm de diâmetro, determine as coordenadas da bola nos vários fotogramas em metros.

Passo 6: Obtenha o valor de y em função do tempo real sabendo que o tempo entre cada imagem é de `dt=1/framerate`.

Tarefas

Use a informação do vídeo e interpolação polinomial para estimar a aceleração da gravidade usando 2 métodos diferentes. Um dos métodos pode usar uma relação entre o tempo entre ressaltos e a altura máxima atingida. O outro método pode usar uma regressão polinomial das ordenadas do movimento da bola durante um ressalto. Use vários ressaltos para estimar g através destes métodos.

PARTE 2
Simulação Computacional da queda da bola

Neste trabalho pretende-se também simular os ressaltos da bola. Para tal, estime como varia a altura máxima atingida nos vários ressaltos e com base nela, estabeleça como varia a velocidade inicial nos vários ressaltos. Resolva depois, numericamente, as equações diferenciais do movimento entre cada embate. Para tal use o método de Euler e um dos outros métodos no anexo 2 para o fazer.

PARTE 3 e 4
Criação de um GUI e elaboração do relatório

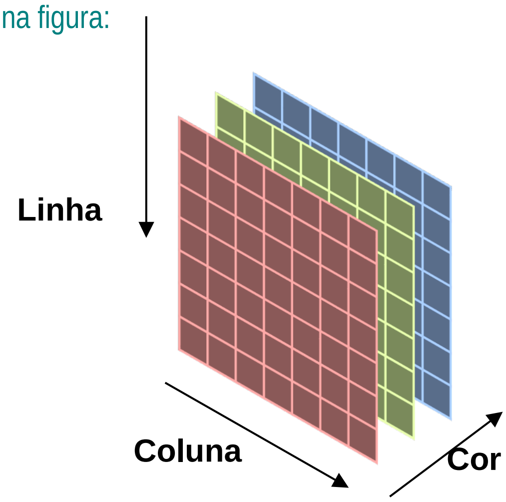
Este trabalho pode beneficiar com a elaboração de um GUI. Desenvolva um GUI que acompanhe este trabalho, bem como um relatório.

O relatório deve:

- discutir o desafio que o trabalho representou e que estratégias desenvolveu para o resolver.
- que conclusões pode tirar do seu estudo (não esquecer: como estimou a aceleração da gravidade; que métodos de interpolação testou e qual o mais adequado ao seu estudo e porquê; como simulou o movimento e como compara os métodos de resolução das equações diferenciais).
- como funciona o seu GUI e qual o propósito que lhe destinou.

ANEXO 1

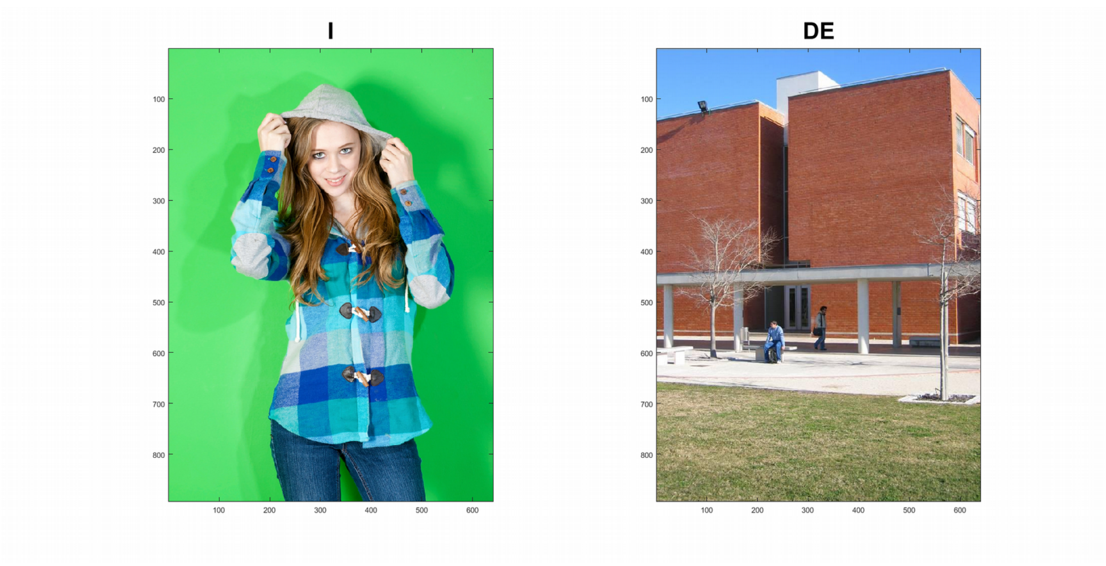
Neste anexo será explicado como podem ser manipuladas imagens e vídeos no Matlab.
Uma imagem, a preto e branco, consiste numa matriz cujo valor de cada elemento indica a "intensidade" de branco num pixel. Já uma imagem a cores é constituída por 3 destas matrizes, uma para cada canal de cor fundamental: Vermelho, Verde e Azul (RGB). Estas 3 matrizes estão concatenadas numa terceira dimensão, fazendo uma matriz tri-dimensional, como exemplificada na figura:



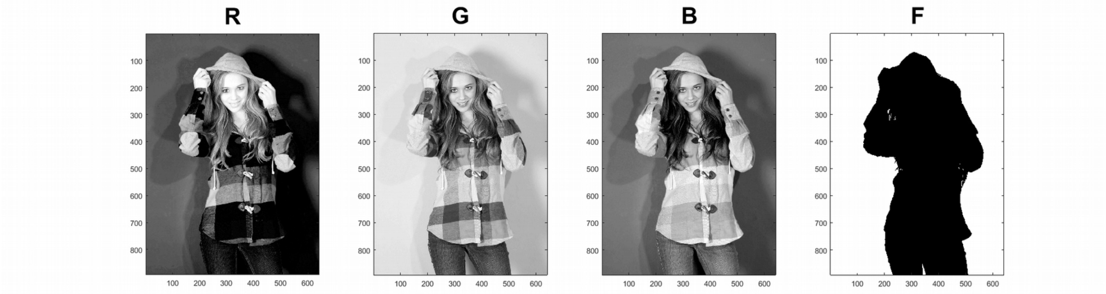
Da mesma forma que para um vetor ou matriz 2D, pode aceder aos elementos de uma matriz 3D através de indexação. Por exemplo, para obter a camada vermelha da imagem, sendo I a matriz de imagem, deve fazer `I(:,:,1)` (1 para o vermelho, 2 para o verde e 3 para o azul). Para selecionar uma secção da imagem, com todas as cores, pode fazer, como para qualquer outra matriz, `I(20:30,10:50,:)` (os valores usados são um exemplo).

Os valores de uma matriz com uma imagem são tipicamente inteiros entre 0 e 255 tipo (uint8) ou reais entre 0 e 1 (double). Para mostrar uma imagem, a função mais prática é o imagesc, já que faz a escala automática ao tipo de dados.

Além de indexação numérica, pode utilizar ainda indexação lógica para editar apenas partes de uma imagem. Por exemplo, considere as seguintes imagens, I e DE:



Se o objetivo for colocar o fundo da imagem DE atrás da rapariga da Imagem I, é preciso identificar as regiões do fundo da imagem I. Para tal deveremos focar-nos nos valores dos canais vermelho `R=I(:,:,1)`, verde `G=I(:,:,2)`e azul `G=I(:,:,3)`da imagem I.



Depois de experimentar vários valores, chega-se à conclusão que os pixéis do fundo verde, são aqueles em que as seguintes condições são verdadeiras: `R<125 & G>140 & B < 145`.

Assim, a matriz F, `F=R<125 & G>140 & B < 145` é uma matriz lógica, de 1's e 0's, sendo os valores a 1 onde as condições são verdadeiras, ou seja, a branco na imagem.

Ao colocar uma matriz lógica ao invés dos índices absolutos ao chamar uma matriz, por exemplo, `I(F)`, obtemos os valores de I em que a condição F é verdade. Por exemplo, para tornar o fundo da camada G da imagem I preto, teremos de fazer, `G(F)=0` .Neste trabalho não irá precisar de manipular mais do que um canal de uma imagem, mas para referência futura, se quisesse tornar o fundo de toda a imagem I a preto teria de usar `I(repmat(F,1,1,3))=0`, de forma a ter também uma matriz lógica tridimensional.

Para colocar o fundo DE na imagem I é só fazer alterar o código para `I(repmat(F,1,1,3))=DE(repmat(F,1,1,3))`

Passando para os vídeos, estes nada mais são que uma matriz a 4D dimensões, em que a quarta dimensão é o tempo. Ou seja, sendo V uma matriz com um vídeo, `V(:,:,:,n)` é a enésima imagem do vídeo (ou *frame*).

Linha **Coluna** **Cor** ***frame***

Outra propriedade importante de um vídeo, é a *framerate* que corresponde ao número de imagens por segundo que o vídeo possui, sendo o seu inverso o tempo entre imagens.

NOTA: há formas de armazenar imagens e vídeos alternativas às que foram expostas neste documento.

ANEXO 2

Para resolver a equação diferencial:

$$m\frac{d^2x}{dt^2}=F(x,t)$$

podem-se usar diversos métodos.

Euler

$$v(t+\Delta t)=v(t)+\Delta t\times\frac{F(x,t)}{m}$$

$$x(t+\Delta t)=x(t)+\Delta t\times v(t)$$

Euler-Cromer

$$v(t+\Delta t)=v(t)+\Delta t\times\frac{F(x,t)}{m}$$

$$x(t+\Delta t)=x(t)+\Delta t\times v(t+\Delta t)$$

Leap-Frog

$$v(t+\Delta t/2)=v(t-\Delta t/2)+\Delta t\times\frac{F(x,t)}{m}$$

$$x(t+\Delta t)=x(t)+\Delta t\times v\left(t+\frac{\Delta t}{2}\right)$$

Verlet

$$x(t+\Delta t)=2x(t)-x(t-\Delta t)+\Delta t^2\times\frac{F(x,t)}{m}$$

$$v(t)=\frac{x(t+\Delta t)-x(t-\Delta t)}{2\Delta t}$$