

Indique o que se entende por quantificação (quantization) num contexto de captura e digitalização de imagem. Indique ainda como este processo determina a qualidade da imagem. Justifique a sua resposta.

Entende-se por quantificação, a atribuição de espaço em memória para guardar informação de uma amostra. Regra geral, a quantificação responde à pergunta “Quantos níveis pode representar cada amostra?”. Em particular, num contexto de captura e digitalização da imagem, quantos mais bits forem reservados para uma amostra, maior a profundidade (channel depth) do canal correspondente. Assim, a quantidade de informação por pixel (após rasterização da imagem) deve ser tanto maior quanto o detalhe de cada amostra, o que se traduz em mais qualidade na resolução do produto final.

Quais as possibilidades de conversão de imagem vectorial para bitmap? E de bitmap para vectorial? Justifique a sua resposta, indicando os métodos de conversão e se há perda de informação no processo.

A conversão de imagem vectorial para bitmap é possível e é denominada por rasterização sendo comumente utilizada para a saída em vídeo ou impressão. O inverso também é possível e ao processo correspondente dá-se o nome de vectorização, sendo sobretudo utilizado para obter imagens escaláveis que podem sofrer ampliação e impressoras em *plotters* sem perda de definição de imagem. Uma vez que estes dois tipos de imagem digital apresentam propriedades bastante distintas, é normal haver perda de informação na conversão entre formatos, nomeadamente no que respeita o processo de rasterização. Uma imagem vectorial pode ser definida como uma descrição paramétrica do conteúdo dessa imagem, onde os objectos 2D ou 3D são representados como modelos. Por tratar-se de uma descrição matemática, este modelo pode sofrer correcções, guardando para cada alteração a sua informação estrutural e respectivo conteúdo semântico. Em contrapartida, um mapa de bits guarda apenas uma matriz de pixels que faz a correspondência bit-a-bit da informação de cor e transparência de cada pixel da imagem com uma posição na matriz, não preservando qualquer informação estrutural ou conteúdo semântico adicionais. Existe, portanto, perda de informação no processo de conversão de imagem vectorial para bitmap e não o contrário.

Considere uma gravação áudio de um discurso. Quando foi capturada esta gravação, a frequência de amostragem utilizada foi de 30KHz. É possível reproduzir fielmente o áudio original? Justifique a sua resposta.

A frequência de amostragem corresponde à taxa de amostras por intervalo que são retiradas de um sinal, para que através da sua análise seja possível recriar o sinal digital definindo apenas alguns valores estratégicos que se vão repetindo até formarem uma sinusoidal. Em particular, o **teorema da amostragem** diz-nos que a frequência de amostragem deve ser duas vezes superior à frequência das componentes do sinal original, garantindo que este possa ser representado sem perdas significativas. O não cumprimento desta regra, leva à corrupção parcial ou total do ficheiro fonte, uma vez que por falta de amostras este não pode ser reconstituído. No que diz respeito à gravação áudio de um discurso, sabe-se que a qualidade para a voz é razoável se a frequência usada rondar os 11KHz, subindo para muito boa com 22KHz e atingindo a excelência aos 44KHz. Assim, tendo sido usada uma frequência de amostragem de 30KHz e assumindo uma quantificação de 16 bits em stereo é possível reproduzir fielmente o áudio original.

Indique como pode usar a informação contida no gráfico seguinte de amplitude em função da frequência para aumentar a compressão de um sinal áudio. Justifique a sua resposta.

A compressão de um sinal áudio pode ser realizada segundo diferentes métodos, dos quais se destacam o **nível mínimo/máximo de audição**, o **efeito de máscara**, o **reservatório de bytes**, **Joint Stereo** e **codificação de Huffman**. Neste exemplo em concreto, interessamos aplicar a compressão perceptual por nível mínimo e máximo de audição, onde à digitalização em bruto do sinal áudio (formato WAV), se eliminam as frequências não audíveis, bem como aquelas que se situam além do limiar da dor, agrupando-se as restantes no ficheiro final comprimido. Pela análise cuidada do gráfico da amplitude em função da frequência, verificamos que só as frequências cuja intensidade se situa entre 0 e 120 décibéis merecem ser preservadas no ficheiro comprimido, pois todas as outras ocuparão espaço desnecessário em memória, uma vez que não são audíveis pelo ser humano.

A produção de DVD-Vídeo inclui a utilização de diversos formatos e normas para informação multimédia. Indique dois destes formatos, as suas características principais e a forma como são usados no DVD-Vídeo.

A produção de DVD-Vídeo inclui a utilização de diversos formatos e normas para informação multimédia, pelo que os ambientes de autoria de DVD-Vídeo devem suportar codificação e compressão de vídeo MPEG-2, codificação de áudio seja por MPEG Layer-III, Dolby Digital ou PCM, legendagem em formato RLE e programas de teste do disco. Codificação de áudio, uma vez que é essencial que um vídeo tenha áudio associado e leitura de RLE, caso seja necessária a legendagem do filme, isto pressupondo que o conteúdo multimédia é feito tendo em conta a sua visualização por audiências que compreendam diferentes idiomas.

Os modelos de cor existentes respondem a diferentes necessidades de aplicação. No caso da compressão de vídeo e imagem indique como podem ser usados de forma distinta os modelos YUV e RGB. Justifique a sua resposta.

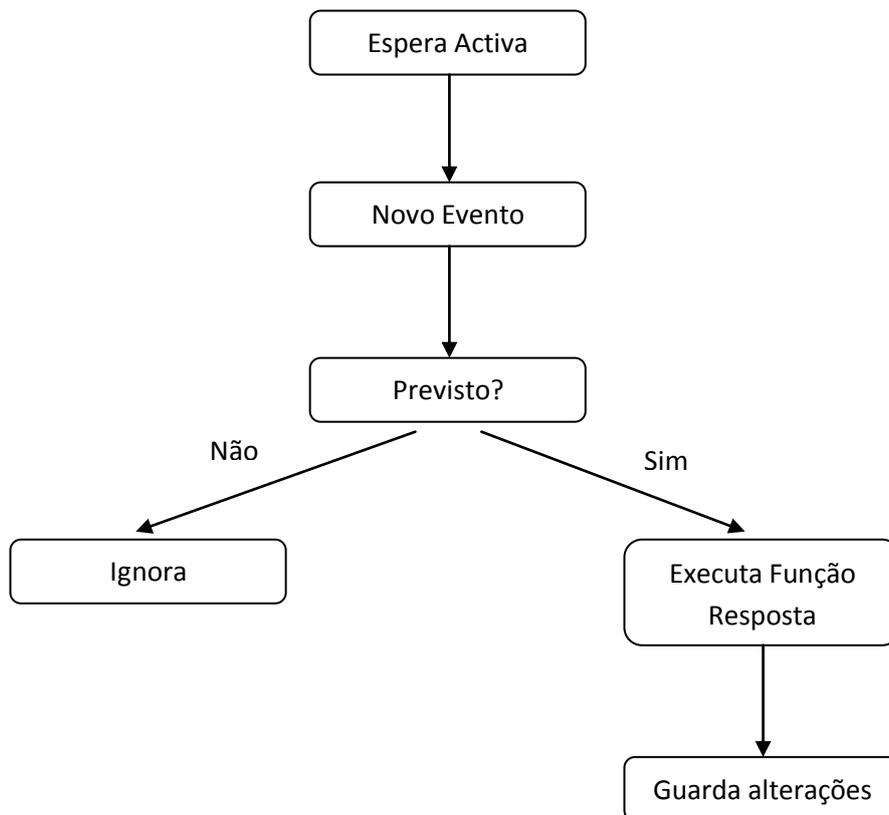
Uma propriedade do modelo de cor YUV que permite aplicá-lo de forma diferente do modelo RGB é a separação entre luminância e cor que caracteriza o primeiro modelo. Sendo o olho humano mais sensível à luz do que à cor, pode poupar-se largura de banda, sem que haja perda de qualidade significativa e que usando o segundo modelo seria desperdiçada. De notar que é possível converter valores entre estes dois modelos, através da fórmula:

$$Y = 0,2125 * R + 0,7154 * G + 0,0721 * B ; U = B - Y ; V = R - Y$$

B = blue (azul), G = green (verde), R = red (vermelho) ;

Y = luminância, U = crominância das linhas, V = crominância das amostras.

Descreva de forma resumida o modelo de eventos do Flash, indicando as fases em que se divide o processamento.



Escreva código AS3 que permita tornar transparentes os pixels das duas cores mais escuras de uma imagem. Assuma que o objecto da classe BitmapData já está preenchido e que para cada pixel as componentes R, G e B são idênticas. Resolva as possíveis ambiguidades da forma que achar mais conveniente, justificando as escolhas. Pode usar os métodos:

```
public getPixel(x:Number, y:Number) : Number  
public setPixel32(x:Number, y:Number, color:Number) : void
```

```
import flash.display.BitmapData;  
  
var theImage:BitmapData = BitmapData.loadBitmap("image");  
  
function brightnessOf(colour:Number) : Number  
    var R:Number = 0;  
    var G:Number = 0;  
    var B:Number = 0;  
  
    R += colour >> 16 & 0xFF;  
    G += colour >> 8 & 0xFF;  
    B += colour >> 0 & 0xFF;  
  
    Var br = Math.sqrt(R * R * .241 + G * G * .691 + B * B * .068);  
    Return br;  
  
Var darkest:Number = 256; // máximo seguro - fora do domínio RGB a 8 bits  
For(var i = 0; i < theImage.width; i++)  
    For(var j = 0; j < theImage.height; j++)  
        Var aux = brightnessOf(theImage.getPixel(i, j));  
        If(aux < darkest) darkest = aux;  
  
For(var i = 0; i < theImage.width; i++)  
    For(var j = 0; j < theImage.height; j++)  
        If(theImage.getPixel(i, j) == darkest) theImage.setPixel32(i, j, 0x00000000);
```

Das orientações para desenvolvimento de aplicações abordadas (por exemplo as de Shneiderman) escolha duas e indique como devem ser aplicadas (para exemplificar pode usar o segundo projecto prático).

As orientações para desenvolvimento de aplicações multimédia recomendadas por Shneiderman incluem regras como a consistência, a utilização de atalhos por parte de utilizadores frequentes, a possibilidade de desfazer uma ou mais acções, a existência de soluções simples perante os erros, ou ainda, acesso rápido aos conteúdos multimédia em memória. A consistência, em particular, pressupõe a padronização de janelas de conteúdos com o mesmo objectivo, facilitando assim ao utilizador a compreensão de novos

ambientes que vão surgindo. Por exemplo, no Adobe Flash, independentemente do ficheiro .as que estivesse correntemente aberto, a interface de programação em ActionScript era idêntica e caracterizava-se, por omissão, por uma página em branco, editável, na qual era sempre possível escrever com a mesma fonte, cor e tamanho de letra. A resposta a eventos imprevisto, ou lançamento de excepções também deve ser o mais clara e intuitiva possível, apresentando-se ao utilizador mensagens com a descrição informal do erro, código (para consulta online) e sempre que possível a sua localização. Tal como a consistência, também este parâmetro foi tido em conta no desenvolvimento do AdobeFlash, uma vez que sempre que era lançada uma excepção, era apresentada na consola a informação da linha onde o erro se encontrava, permitindo ao programador multimédia limitar o espaço de procura, o que é extremamente importante em projectos com grande dimensão.