universidade de aveiro

Codificação de Áudio e Vídeo
Projeto 1 - Manipulação de Áudio e Vídeo
DETI - Universidade de Aveiro

Autores:
David Almeida - 80301
João Cruz - 76517
Renato Fontão - 80286
Grupo:
4
Docente:
António Neves
Data:
18/10/2019

Introdução

O projecto surge no âmbito da Unidade Curricular CAV (Codificação de Áudio e Vídeo), cujo o objectivo é a ambientação à manipulação de ficheiros áudio e imagem/vídeo. Para desenvolver os problemas apresentados, utilizou-se como linguagem de programação python e bibliotecas da linguagem.

Parte I

Cópia sample-by-sample de amostra áudio

Com recurso às bibliotecas python wave e struct, foi lido o ficheiro .WAV pretendido e obteve-se todas as samples áudio. De seguida, foi feito o unpack do canal esquerdo e direito do array que continha as samples e o pack para o novo array de samples áudio. Por fim, escreve-se num novo ficheiro os valores copiados.

Cópia sample-by-sample da imagem/vídeo

Com recurso ao opencv, importaram-se os vários frames da imagem/vídeo correspondente. Para fazer a cópia sample a sample corremos todos os pixels da foto e copiamos o seu valor para uma nova frame, de igual tamanho à original. Finalmente, gravou-se a nova frame como imagem.

Display de imagem/vídeo no ecrã

Neste caso, é dado como argumento de entrada o nome do ficheiro que se pretende abrir e um valor 0/1, que corresponde ao tipo de ficheiro. Recorrendo à função “imshow” do OpenCV mostra-se no ecrã a imagem/vídeo.

Calcular o Histograma de um ficheiro áudio

Tal como na cópia sample a sample, recorreu-se à biblioteca wave e struct para ler as samples e separar os canais, assim como calcular a sua média. Posteriormente, com recurso à biblioteca matplotlib, foi feito o plot com o histograma dos canais, esquerdo, direito e mono.

Calcular o Histograma de uma imagem/vídeo

Tal como para os ficheiros áudio, recorreu-se à biblioteca matplotlib para fazer o histograma. Para o caso da imagem os histogramas apresentados são das cores R, G, B e Gray.

Reduzir o número de bits a cada sample áudio

Para a execução deste programa, é preciso apresentar como argumento de entrada o nome do ficheiro áudio, presente numa das pastas de ficheiros áudio, e o número de bits com o qual se pretende quantizar cada sample do áudio.

O fluxo do programa consiste em definir os intervalos de quantização, de acordo com o número de bits e o mínimo e máximos do sinal, e redefinir os valores, para coincidir com os valores dos níveis de quantização.

Reduzir o número de bits em cada sample de imagem/vídeo

A abordagem utilizada para a redução do número de bits das imagens foi utilizada uma quantização não uniforme baseada no algoritmo de Lloyd, apresentado nos slides da aula 3.

No caso do vídeo, foi utilizada a mesma abordagem da imagem, aplicando para cada frame. No entanto, o tempo de processamento é demasiado elevado se o utilizador quiser várias iterações, que seria o ideal para o algoritmo convergir.

SNR e máximo erro absoluto de um ficheiro áudio em comparação com o original

Comparou-se o sinal quantizado na alínea anterior e de acordo com a fórmula $SNR = 10.log_{10}(\frac{\sigma_x^2}{\sigma_q^2})$, para tal fizemos a soma dos quadrados da diferença do ficheiro original para o quantizado, e a soma dos quadrados do original, aplicando a fórmula anterior para obter o valor em dB do SNR. Para verificar qual o máximo do erro absoluto basta fazer

a diferença entre as samples do ficheiro original e quantizado e verificar qual a maior diferença que obtemos.

SNR e máximo erro absoluto de uma imagem/vídeo em comparação com o original

O método utilizado para calcular o SNR e o máximo erro absoluto assemelha-se ao utilizado no áudio, a no caso da imagem foi feita a comparação pixel a pixel.

Parte II

Cálculo baseado num método de estados finitos da entropia de um ficheiro

Para o cálculo da entropia de um ficheiro de áudio, imagem ou vídeo, foi utilizado o método de de estados finitos baseado em modelos de Markov de ordem. Para cada bit do ficheiro, foi a analisada a probabilidade do próximo bit ser ‘1’ ou ‘0’. Ou seja, considerou-se o ficheiro como um array de bits.

De seguida, foram aplicadas as equações de cálculo das probabilidades de cada bit através das probabilidades condicionais apresentadas nas aulas. Finalmente calculou-se a entropia do sistema.

Verificamos que os ficheiros de áudio possuíam o valor mais elevado de entropia dos três tipos de ficheiro e que os ficheiros de vídeo possuíam o valor mais reduzido. Concluímos que este fenómeno se deve ao facto de que, por norma, ficheiros com um número mais elevado de bits tendem a ter valores de entropia mais baixos.

Conclusão:

Cremos que a realização deste projeto cumpriu os objetivos pretendidos, nomeadamente a familiarização com alguns métodos de manipulação e análise de ficheiros de áudio, imagem e vídeo. Permitiu-nos também aplicar vários conceitos teóricos, como a quantização de um ficheiro, o respetivo cálculo da SNR e entropia.

No geral, consideramos que o projeto permitiu aprofundar os conceitos abordados nas aulas e conferiu-nos ferramentas que serão essenciais na realização dos próximos projetos.