

Universidade da Beira Interior

Departamento de Informática



**Departamento de
Informática**

Nº 1 - 2021: *Projeto de Multimédia*

Elaborado por:

Nº 45609 - Miguel Pereira
Nº 45894 - Guilherme Nunes

Orientador:

Professor/a Doutor/a João Dias

16 de janeiro de 2022

Conteúdo

| | |
|---|-----------|
| Conteúdo | 0 |
| 1 Introdução | 1 |
| 2 Algoritmos | 3 |
| 2.1 Imagens | 3 |
| 2.1.1 Preto e Branco(Black And White) | 3 |
| 2.1.2 Negativo (Negative) | 3 |
| 2.1.3 Soma (Add) | 3 |
| 2.1.4 Sobreposição (Overlay) | 3 |
| 2.1.5 Ou (Or) | 4 |
| 2.2 Audio | 4 |
| 2.2.1 Soma (Add) | 4 |
| 2.2.2 Sobreposição (Overlay) | 4 |
| 2.2.3 Lowpass Highpass | 4 |
| 2.3 Vídeo | 5 |
| 2.3.1 Guardar Frames (Save Frames) | 5 |
| 2.4 Compressão | 5 |
| 2.4.1 Redimensionar (Resize) | 5 |
| 2.4.2 Corte (Cropping) | 5 |
| 2.4.3 Comprimir (Compress) | 5 |
| 3 Resultados | 7 |
| 3.1 Imagens Originais | 7 |
| 3.2 Imagens Modificadas | 8 |
| 4 Conclusão | 13 |
| 5 Bibliografia | 15 |

Capítulo

1

Introdução

Este projeto , lançado pelo Professor Doutor João Dias, foi feito no âmbito da Unidade Curricular de Multimédia. O seu objetivo era que os alunos pudessem em prática vários conhecimentos adquiridos ao longo do semestre e criassem algoritmos que permitissem efetuar operações, compressões e filtrações a diversos tipos de multimédia como por exemplo a imagens, vídeo e áudio.

Este projeto foi realizado com a linguagem de programação Python e o IDE utilizado foi o PyCharm. A utilização da linguagem foi uma sugestão do professor ainda que tenha dado liberdade aos alunos para realizarem o trabalho naquela que preferissem.

A divisão do trabalho entre os dois elementos do grupo foi a seguinte. O Guilherme Nunes ficou responsável pela criação dos filtros de imagem , do vídeo e da compressão, já o Miguel Pereira ficou responsável pela criação das operações de imagem e do áudio, o relatório foi feito pelos dois alunos.

Decidimos criar um pequeno menu, não que fosse necessário ou algo a ter em conta na avaliação mas achamos que para questões de apresentação e organização poderia ser útil e interessante.

Capítulo

2

Algoritmos

2.1 Imagens

2.1.1 Preto e Branco(Black And White)

O filtro Black And White abre a imagem e percorre-a através de um ciclo "for", multiplicamos os pixels pelo valor dado para ficar com aquele tom.

2.1.2 Negativo (Negative)

No filtro de negativo o valor de cada pixel da matriz é retirado a 255, e o valor resultante substitui-o, ou seja, o valor original do pixel é substituído pelo valor que lhe falta para chegar a 255 que é a sua cor inversa, isto está incluído num ciclo for que corre toda a matriz para que este procedimento aconteça para todos os pixels.

2.1.3 Soma (Add)

Na operação de adição soma-se o valor compreendido entre 0 e 255 dos pixels das duas imagens e a imagem resultante vai ter o valor do somatório desses 2 naquele pixel, se o valor for superior a 255 ele irá ficar como 255. Este procedimento está incluído num ciclo for que corre toda a matriz para que este procedimento aconteça para todos os pixels.

2.1.4 Sobreposição (Overlay)

A operação de Overlay coloca uma imagem sobre outra, está a cargo do utilizador escolher o local onde a pretende colocar, no nosso caso escolhemos a

posição (0,0), ou seja, na primeira posição da matriz, canto superior esquerdo.

2.1.5 Ou (Or)

Na função "Or" das matrizes das imagens são corridas através de um ciclo "for" e são analisados todos os pixels destas, seguida é feita uma seleção entre os pixels de posições iguais das imagens, aquele que tiver um valor entre 0 e 255 maior será o selecionado e ocupará o seu lugar na imagem que será formada, após a exportação teremos uma imagem em que foram selecionados os pixels com maior valor para as respectivas posições.

2.2 Audio

2.2.1 Soma (Add)

A soma de audios funciona da seguinte forma, os dois audios são "chamados" para que os possamos utilizar, criamos uma variável para o novo som e fazemos $som1 + som2 = novosom$ e por fim o som é exportado, o que vai acontecer é que vamos criar um ficheiro de audio em que é tocado o primeiro som e de seguida sem intervalos irá tocar o som 2.

2.2.2 Sobreposição (Overlay)

O Overlay de audios funciona de forma bastante similar só que ao invés de tocar um audio após o outro estes começam ao mesmo tempo e sobrepõem-se, assim criando um novo audio que é uma mistura dos dois, para criar este algoritmo os dois audios foram "chamados" para o programa, criou-se uma variável para armazenar o novo som, fizemos $som1 * som2 = novosom$ e exportou-se o novo audio.

2.2.3 Lowpass Highpass

O Lowpass e o Highpass transformam as frequências dos audios subindo ou descendo-as, decidimos usar as bibliotecas pois não encontramos outra solução.

2.3 Vídeo

2.3.1 Guardar Frames (Save Frames)

O save frames começa por abrir o vídeo e através de um ciclo "for" e de um "while" corremos os frames todos e guardamos com a numeração:("%dframe.jpg") , sendo adicionado 1 sempre que corre, guardam-se os mesmos numa pasta frames, de seguida armazenamos os frames todos da pasta numa lista e através de outro ciclo fazemos o preto e branco dado na aula, enquanto isso os frames transformados estão a ser guardados numa pasta bwframes, de seguida como os frames (11,22,33,44,,55...) tinham erro na orientação decidimos retirá-los pois não afetava o vídeo final decidimos passar para uma pasta trash, através de um algoritmo de sorting encontrado na internet, conseguimos juntar os frames todos de volta através de um ciclo "for" e o write da biblioteca opencv.

2.4 Compressão

2.4.1 Redimensionar (Resize)

No resize começamos por ler a imagem e retirar os seus atributos, de seguida retiramos 3 os planos RGB e são separados, esses valores serão reduzidos para metade, fazendo com que a imagem sofra um redimensionamento para metade do tamanho. Assim comprimindo-a.

2.4.2 Corte (Cropping)

No cropping é executado um procedimento parecido ao descrito no resize, porém desta vez utilizamos um ciclo "for" para alterar o comprimento e altura, utilizando certos pixels fazemos com que apenas parte da imagem apareça no plano final.

2.4.3 Comprimir (Compress)

A função Compress dá uso aos parâmetros disponíveis no "SAVE" para otimizar a imagem, criamos um ciclo "for" para fazer isso com todas elas.

Capítulo

3

Resultados

3.1 Imagens Originais



Figura 3.1: Imagem 1



Figura 3.2: Imagem 2



Figura 3.3: Imagem 3

3.2 Imagens Modificadas



Figura 3.4: Adição

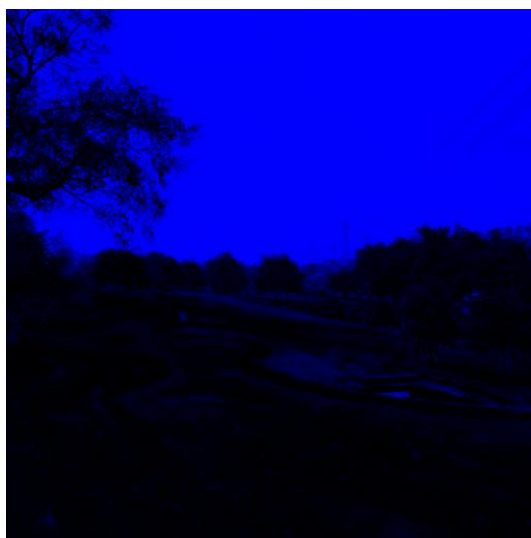


Figura 3.5: Blue Filter



Figura 3.6: Black And White

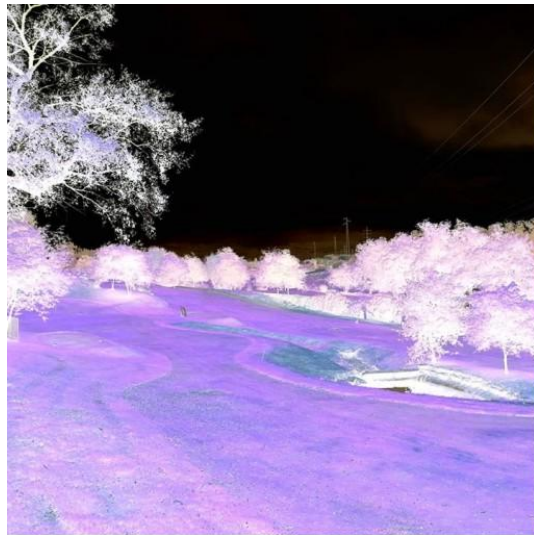


Figura 3.7: Negativo



Figura 3.8: Ou



Figura 3.9: Subtração

Capítulo

4

Conclusão

Para além de elemento de avaliação este trabalho fez-nos perceber o que está por detrás de uma modificação a um elemento multimédia, muitas vezes, através de aplicações de edição de vídeo ou de audio este tipo de operações e filtros estão simplificados por um botão que automaticamente efetua as alterações por nós mas com este trabalho foi possível perceber a estrutura dos algoritmos responsáveis por essas alterações e perceber ainda o que realmente acontece, como a imagem é convertida em matriz, como o audio é passado a binário num Array, etc...

O tipo de atividade praticada neste relatório não é recorrente, como foi dito anteriormente se estiver-mos a utilizar uma aplicação existe uma interface gráfica que facilita este tipo de aplicações e transformações nos elementos multimédia. Quando se escreve código existem bibliotecas que facilitam este tipo de trabalho e resumem várias linhas numa única função. Porém o entendimento do procedimento facilita a sua utilização e o conhecimento do método de aplicação pode ser útil numa outra circunstância.

Capítulo

5

Bibliografía

<https://www.tutorialspoint.com/read-and-write-wav-files-using-python-wave>

<https://docs.python.org/3/library/wave.html>

<https://stackoverflow.com/en/question/3TKqf>

<https://www.youtube.com/watch?v=krLq51WiGmUab> *channel = HowTo*

<https://gist.github.com/gchavez2/53148cdf7490ad62699385791816b1ea>

<https://stackoverflow.com/questions/10465747/how-to-create-a-white-image-in-python>

<https://www.youtube.com/watch?v=WK8q4z1HBHMab> *channel = LeonMarsdenYouTube*

<https://www.youtube.com/watch?v=QPLYTFgTQIgab> *channel = DataCamp*

<https://stackoverflow.com/questions/33311153/python-extracting-and-saving-video-frames>

<https://stackoverflow.com/questions/44947505/how-to-make-a-movie-out-of-images-in-python>

<https://www.includehelp.com/python/resize-an-rgb-image-without-using-any-inbuilt-functions.aspx>

<https://stackoverflow.com/questions/65911324/blue-image-filter-in-python>

<https://stackoverflow.com/questions/58853388/how-do-i-preserve-the-order-of-images-while-creating-video-in-python>

<https://www.geeksforgeeks.org/image-processing-without-opencv-python/>