

Faculdade de Engenharia da Universidade do Porto



Deteção de Cortes de Cena e Estimação de Movimento

Tomas Sousa Pereira & Joaquim Pedro Santos

Abril 2019

Índice

1	Introdução	1
2	Comparar Histogramas de cor	2
3	Comparar Image Activity Measure	3
4	Estimação de movimento	4
5	Conclusão	6

1. Introdução

Neste trabalho fomos propostos a explorar técnicas de detecção de cenas e estimação do movimento usando o Matlab como tal foram nos introduzidos os seguintes conceitos.

Dentro dos algoritmos propostos decidimos implementar: comparação de histogramas de cor, medida da ativade de uma imagem e por fim implementamos uma pesquisa rápida baseada no algoritmo de 3 step search.

Estamos assim preparados para começarmos a abordar este trabalho. Como previsto todo o código será anexado, para o correr será preciso escolher o vídeo para serem detetados as cenas, e de seguida escolher um dos 3 métodos. Neste trabalho os resultados são relativos ao vídeo: wg_cs_1.mpg.

Para cada um dos métodos é criado um ficheiro scenes_nome_do_metodo.txt onde está representado o índice dos frames correspondentes ao inicio de cena, também é criado um ficheiro scenes_nome_do_metodo.mp4 onde estão presentes as mesmas cenas representadas num vídeo.

2. Comparar Histogramas de cor

O primeiro método desenvolvido foi um método que conceptualmente nos pareceu bastante simples e lógico. A ideia seria analisar cada frame do vídeo, e comparar as cores deste frame com os frames subsequentes e verificar se existiam semelhanças. Um dos problemas que encontramos foi quantificar estas semelhanças, o método que escolhermos foi calcular o RMSE (Root Mean Square Error) e definimos uma threshold experimentalmente.

Numa primeira abordagem este método parecia-nos poder obter bons resultados mas ao correr o programa percebemos que com as cores parecidas este método parecia de eficácia. O maior erro foi nas imagens de transição onde as cores já começavam a alterar no entanto a cena ainda não tinha alterado, como podemos ver na Figura 2.1, a primeira e a terceira fotografias são cenas diferentes, no entanto a segunda imagem ,que foi identificada como uma cena devido a dessemelhança de cores, não é considerada uma cena.



Figura 2.1: Comparação de Frames de cenas diferentes

Embora algum problema nas zonas de transacção o algoritmo até obteve bons resultados na Figura 2.2 seguinte podemos observar o ficheiro scenes_histograma.txt onde encontramos os índices e algumas contagens.

1, 23, 24, 79, 80, 100, 121, 127, 132, 133, 160, 207, 223, 229, 238, 239,
240, 248, 257, 273, 286, 287, 301, 307, 335, 338, 358, 359, 382, 388, 392,
393, 406, 449
Total scenes: 34
Total frames: 489

Figura 2.2: Ficheiro scenes_histograma.txt

3. Comparar Image Activity Measure

Após compararmos frames utilizando os histogramas de cor, decidimos procurar alguma ferramenta mais eficaz. Para tal como sugerido pela Professora, utilizamos a Image Activity Measure (IAM). Esta medida fornece uma medida da complexidade ou actividade espacial da imagem e é calculada com a seguinte expressão.

$$IAM = \frac{1}{N * M} * \left(\sum_{i=1}^{M-1} \sum_{j=1}^N |I(i, j) - I(i + 1, j)| + \sum_{i=1}^M \sum_{j=1}^{N-1} |I(i, j) - I(i, j + 1)| \right) \quad (3.1)$$

Com esta medida, imagens com IAM semelhantes são imagens semelhantes, por isso mais uma vez foi encontrado um threshold que permitiu ter resultados bastante razoáveis. No entanto dentro da nossa gama de resultados, encontramos um pequeno erro desta medida. Como esta calcula o nível de atividade na imagen com diferente nível de atividade considera cenas diferentes. No entanto como podemos observar na figura 1.3 o que o algoritmo identifica como três cenas diferentes o olho humano apenas identifica uma cena.



(a) 1^a Imagem

(b) 2^a Imagem

(c) 3^a Imagem

Figura 3.1: Comparaçao de Frames de cenas diferentes

Embora tenha acontecido este pequeno erro, o nosso programa conseguiu identificar 28 cenas em 489 frames como demonstrado na Figura seguinte.

```
1, 23, 24, 79, 80, 100, 121, 127, 132, 133, 160, 207, 223, 229, 238, 239,  
240, 248, 257, 273, 286, 287, 301, 307, 335, 338, 358, 359, 382, 388, 392,  
393, 406, 449  
Total scenes: 34  
Total frames: 489
```

Figura 3.2: Ficheiro scenes_iam.txt

4. Estimação de movimento

Na última parte do trabalho implementamos o 3 Step Search um algoritmo de pesquisa rápida. Este algoritmo é muito utilizado para a compressão de vídeos e é dos algoritmos de eleição para efetuar Block-matching.

O nosso algoritmo funciona da seguinte maneira, o primeiro passo é calcular o vetor movimento entre dois frames consecutivos. Como descrito na especificação do algoritmo, selecionam-se 8 pontos em redor de um ponto inicial, de seguida calcula-se o custo do bloco associado a cada um dos pontos. Este custo pode ser calculado de diversas maneiras no entanto nós utilizamos o Mean Absolute Difference (MAD), esta é uma função que não é muito exigente computacionalmente e muito popular pelos seus resultados. Após calcular os custo volta-se a marcar 8 pontos e encontra-se aquele com menor custo, assim sucessivamente até se obter um passo 1 (a Figura seguinte ajuda uma melhor compreensão).

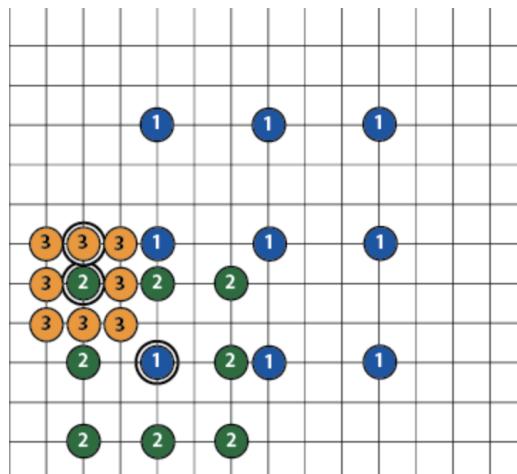


Figura 4.1: Algoritmo 3 Step Search

Uma vez calculado o vetor de movimento aplica-se este vetor no na imagem a ser comparada. De seguida calcula-se o peak signal-to-noise entre a imagem de referência e a imagem a ser comparada depois de adicionado o vetor movimento. Esta medida permite analisar o impacto que o ruído tem numa imagem. Ao efetuar um plot desta medida ao longo dos frames obteve-se o gráfico seguinte .

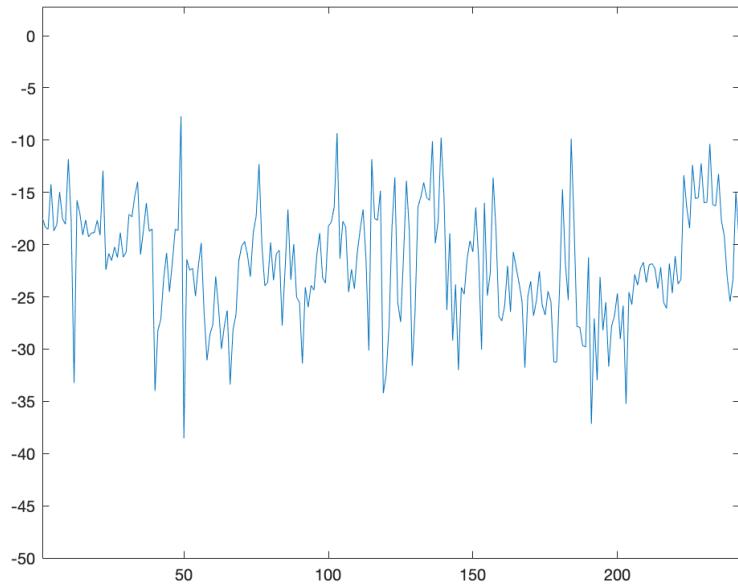


Figura 4.2: Plot do PSNR ao longo dos frames

Com este gráfico procuramos os maiores picos, estes picos correspondem ás maiores diferenças entre frames, por isso podemos considerar que representam cenas diferentes.

Com este algoritmo alcançamos os melhores resultados, como seria de esperar, assim demonstrado na figura seguinte

```
24, 100, 182, 238, 290, 336, 382, 434, 478,  
Total scenes: 9  
Total frames: 489
```

Figura 4.3: Ficheiro scenes_3ss.txt

5. Conclusão

Este trabalho foi extremamente proveitoso e bem conseguido conseguimos aprender e implementar algoritmos de detecção de cenas.

A comparação utilizando os histogramas de cor fornecem bons resultados numa boa perspectiva no entanto são insuficientes para frames cujo os tons de cor sejam semelhantes.

Por outro lado a comparação utilizando a IAM permitiu obter melhores resultados em relação ao método anterior. Especialmente pelo método de comparação das imagens que não mede as características intrínsecas das imagens mas o seu o seu nível de actividade, ainda assim não foi tão bem sucedido como o algoritmo seguinte.

Por último e com os melhores resultados, implementados uma pesquisa rápida 3 Step Search. Este algoritmo como o anterior não mede as características das imagens, a sua maior vantagem é que este método ao calcular o vetor movimento tem conta a imagem seguinte, permitindo assim um melhor seguimento das cenas.

No fim concluimos considerando a Three Step Search o algoritmo com melhor performance, percebendo assim a razão da sua popularidade.