

Identificando transições abruptas em vídeos (trabalho 3)

Introdução ao Processamento de Imagem Digital

Randerson A. Lemos (103897) 2022-1S

1 Introdução

Vídeos são compostos por quadros de mesma dimensão organizados temporalmente. Esses quadros, ao serem apresentados sequencialmente (no contexto do vídeo), podem mostrar transições suaves ou abruptas. Quando presentes, essas transições abruptas entre quadros podem indicar uma sequência significativamente nova de conteúdo no vídeo e, por isso, tais transições, quando identificadas, podem ser utilizadas como material de resumo do próprio vídeo. Neste trabalho, utilizaremos quatro técnicas capazes de identificar transições abruptas entre quadros de um vídeo de acordo com certas métricas de distância. As técnicas utilizadas são: **diferenças entre pixels**; **diferenças entre blocos**; **diferenças entre histogramas**; e **diferenças entre mapas de bordas**.

1.1 Diferenças entre pixels

Aqui, para que dois quadros consecutivos sejam considerados diferentes é feita uma contagem do número de par de pixels (um de cada quadro) que são diferentes de acordo com uma tolerância T_1 . Se essa contagem for superior a um limiar T_2 , os quadros comparados são considerados diferentes e a transição entre eles entendida como abrupta. Tanto a tolerância T_1 quanto o limiar T_2 são parâmetros definidos pelo usuário e apresentam valores ótimos dependentes das características de cada vídeo analisado. Para o cálculo da diferença entre pixels, cujo valor resultante é utilizado na comparação com a tolerância T_1 , utiliza-se a norma 1, isto é, a diferença absoluta entre os níveis de cinza do par de pixels analisados dos quadros consecutivos.

1.2 Diferenças entre blocos

Na diferenças entre blocos é necessário subdividir os quadros consecutivos em blocos (cada quadro com seu conjunto de blocos). Feito isso, os pares de blocos correspondentes entre os quadros consecutivos são comparados e, de acordo com uma tolerância T_1 , são ou não considerados diferentes. O número de pares de blocos entre quadros consecutivos considerados diferentes é comparado a um limiar T_2 e, caso esse número seja superior a tal limiar, os quadros são considerados diferentes. Os blocos que subdividem os quadros possuem ou dimensão de 8×8 ou de 16×16 . Tanto a tolerância T_1 quanto o limiar T_2 são definidos pelo usuário e seus valores ótimos são dependentes das características de cada vídeo analisado.

1.3 Diferenças entre histogramas

Diferente das anteriores, nesta abordagem utilizaremos os histogramas dos quadros consecutivos, e não os valores dos pixels em si, como fonte para realização das comparações e, consequentemente, determinação de transições abruptas ou não entre tais quadros. A diferença D_i entre histogramas de quadros consecutivos i e $i + 1$ de um vídeo é calculada como

$$D_i = \sum_{j=1}^B |H_i(j) - H_{i+1}(j)|,$$

em que B denota o número total de *bins* no histograma e $H_i(j)$ é o valor do histograma para o i -ésimo quadro no nível j . Para avaliar se os quadros consecutivos são diferentes o suficiente para considerar que a transição correspondente seja abrupta, um limiar T é calculado a partir do conjunto de todas as diferenças D_i obtidas. Em posse da média μ e do desvio padrão σ das diferenças D_i o limiar T é obtido pela equação

$$T = \mu + \alpha\sigma,$$

em que α é definido pelo usuário. Ao parâmetro *alpha* é atribuído valores típicos entre 3 e 6."

1.4 Diferenças entre bordas

Na diferenças entre bordas, utiliza-se o mapa de bordas dos quadros consecutivos como fonte de informação para determinação ou não de uma transição abrupta entre esses quadros. A abordagem adotada para verificação da diferença entre quadros consecutivos foi a contagem do número de pixels das bordas de cada quadro seguida da subtração absoluta desses valores. O número resultante dessa operação é comparado com um limiar T . Caso tal número seja superior que esse limiar, os quadros são considerados diferentes e a transição correspondente abrupta.

2 Solução

A solução utiliza a linguagem de programação Python e conta com o auxílio do gerenciador de projetos e pacotes Conda. Assumindo que o usuário tenha o Conda instalado em sua máquina, a configuração do projeto pode ser feita pela execução do comando `conda env create -f environment.yml` a partir da pasta do projeto **trab3**. Esse comando cria o ambiente de trabalho **mc920-trab3** e instala os seguintes módulos: `opencv`, `numpy`, `scipy`, `pandas`, `matplotlib`. Finalizada a configuração do ambiente de trabalho em questão, o usuário deve executar o comando `source source.sh`¹ para carregar as variáveis de ambiente adequadas e, assim, poder usar os programas do projeto dentro do próprio ambiente de trabalho recém configurado.

Dos arquivos presentes na pasta do projeto **trab3**, destacam-se as pastas **assets**, **classes**, **tex** e o programa **main.py**. A pasta **assets** contém vídeos no formato `png` ou `mpg` que podem ser utilizados para a aplicação das técnicas de identificação de transições abruptas implementadas. A pasta **classes** contém arquivos com as implementações das diferentes técnicas de identificação de transições abruptas em formato de classes. Os arquivos das classes das técnicas são: **blocksdifferences.py**, **edgesdifferences.py**, **histogramdifferences.py**, e **pixelsdifferences.py**. A pasta **tex** contém os arquivos `Latex` deste relatório. O programa **main.py** contém as implementações necessárias para aplicar as técnicas de identificação de transições abruptas no vídeo de entrada fornecido pelo usuário. Informações pertinentes das implementações serão apresentadas a seguir.

2.1 Main.py

O programa **main.py** é responsável pela aplicação de todas as técnicas consideradas de identificação de transições abruptas entre quadros consecutivos de vídeos. Para ser executado, esse programa precisa receber o parâmetro de entrada **video_entrada**:

- ao parâmetro **video_entrada** deve-se fornecer o nome do arquivo do vídeo a ser utilizado para aplicação das técnicas de identificação de transições abruptas.

O programa **main.py** disponibiliza tanto um vídeo no formato `mp4` dos quadros considerados como pertencentes a transições abruptas como também gráficos em formato `png` da evolução dos valores de distância utilizados para determinação de transições abruptas ou não. Todo esse conteúdo é salvo dentro da pasta **out** que está localizada dentro da pasta do projeto **trab3**.

Um exemplo de como executar o programa **main.py** utilizando os recursos contidos dentro do próprio projeto é: `python3 main.py -video_entrada=assets/lisa.mpg`.

O programa **main.py** executa quatro funções principais, as quais são: **main_pixels_differences(path, stem)**; **main_blocks_differences(path, stem)**; **main_histog_differences(path, stem)**; e **main_edges_differences(path, stem)**. Cada uma dessas funções é responsável por executar a técnica de identificação de transições abruptas que encapsula várias vezes utilizando diferentes combinações de parâmetros de entrada (tolerâncias e limiares). Essas funções também estão encarregadas de salvar todo o conteúdo gerado (os vídeos e gráficos) com nomes intuitivos dentro da pasta **out**. A seguir olharemos para essas funções e, principalmente, para as técnicas de identificação de transições abruptas associadas.

2.2 main_pixels_differences(path, stem)

Essa função roda a classe **PixelsDifferences** sobre as tuplas resultantes do produto cartesiano entre os conjuntos $MPNDS \times MPNNS$ cujos elementos são

$$MPNDS = \{0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.7\}$$

e

$$MPNNS = \{0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.7\}.$$

¹O comando que configura o ambiente de trabalho `mc920-trab3` precisa ser executado apenas um vez. Assim sendo, depois que este ambiente está configurado, o usuário precisa apenas executar o comando `source source.sh`

MPNDS é sigla de *Maximum Pixel Normalized Distances* e MPNNS é sigla de *Maximum Pixel Normalized Numbers*. Os valores do conjunto MPNDS fazem referência a tolerância T_1 utilizada para verificar se dois pixels são ou não significativamente diferentes. Os valores do conjunto NPNNS fazem referência ao limiar T_2 , que é utilizado para determinar se o número final de pixels considerados diferentes é suficiente para que os quadros consecutivos sejam classificados como tendo uma transição abrupta. Ambos os valores dos conjuntos em questão apresentam domínio pertencente ao intervalo $[0, 1]$ porque tanto o calculo da tolerância T_1 quando do limiar T_2 e feito de maneira normalizada. Para comparação com a tolerância T_1 , o cômputo da distância entre pixels é feita da seguinte maneira

$$d_{xy} = |Q_i(x, y) - Q_{i+1}(x, y)|/255,$$

em que $Q_i(x, y)$ é o valor de intensidade do pixel da posição (x, y) do i -ésimo quadro em escala de cinza do vídeo analisado. Como a diferença máxima entre pixels em escala de cinza representados com 1 bite é de 255, o valor da variável d sempre vai pertencer ao intervalo $[0, 1]$. Para comparação com o limiar T_2 o cômputo do numero de pixels que apresentam transições abruptas é feita assim

$$D_i = \sum_{x=0}^{x=M} \sum_{y=0}^{y=N} (1 \text{ se } d_{xy} > T_1 \text{ caso contrario } 0).$$

Caso D_i , isto é o número de pixels considerados diferentes de acordo com a tolerância T_1 entre o i -ésimo e $(i+1)$ -ésimo quadros, for superior que o limiar T_2 , o quadro $(i+1)$ -ésimo é classificado como contendo uma transição abrupta.

2.3 main_blocks_differences(path, stem)

Esse função roda a classe **BlocksDifferences** sobre as tuplas resultantes do produto cartesiano entre os conjuntos $MBNDS \times MBNNS$ cujos elementos são

$$MBNDS = \{0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.7\}$$

e

$$MBNNS = \{0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.7\}.$$

MBNDS é sigla de *Maximum Block Normalized Distances* e MBNNS é sigla de *Maximum Block Normalized Numbers*. Os valores do conjunto MBNDS fazem referência a tolerância T_1 utilizada para verificar se dois blocos são ou não significativamente diferentes. Os valores do conjunto NBNNS fazem referência ao limiar T_2 , que é utilizado para determinar se o número final de blocos considerados diferentes é suficiente para que os quadros consecutivos sejam classificados como tendo uma transição abrupta. Ambos os valores dos conjuntos em questão apresentam domínio pertencente ao intervalo $[0, 1]$ porque tanto o calculo da tolerância T_1 quando do limiar T_2 e feito de maneira normalizada.

Para comparação com a tolerância T_1 o cômputo da distância entre blocos é feita da seguinte maneira

$$d_{ai} = \sum_{w=0}^8 \sum_{k=0}^8 (B_{a_i}(w, k) - B_{a_{i+1}}(w, k))^2 / 255^2,$$

em que d_{ai} é a distância quadrática normalizada dos a -ésimos blocos entre os quadros i e $i + 1$. Logo $B_{a_i}(w, k)$ é o valor em escala de cinza do pixel que está na posição (w, k) do a -ésimo bloco da i -ésima imagem. Os valores de d_{ai} estão contidos no intervalo $[0, 1]$ (o termo 255^2 é um fator de normalização).

Para comparação com o limiar T_2 , o número total de blocos que violam a tolerância T_1 é levantado e, se esse valor ultrapassar o limiar T_2 , os quadros consecutivos utilizados na comparação por meio dos seus blocos são considerados como possuidores de transições abruptas. Os valore utilizados em T_2 estão normalizados pelo número total de blocos que cabe nos quadros do vídeo analisado e por isso também estão contidos dentro do intervalo $[0, 1]$.

2.3.1 Detalhes de implementação e execução

Como requisitado no trabalho, a classe **BlocksDifferences** é capaz de trabalhar com blocos de 8×8 ou 16×16 . No entanto para os resultados mostrados, apenas execuções com blocos de tamanho 8×8 foram considerados. Sobre o problema do casamento exato ou não entre as dimensões dos blocos (que pode ser de 8×8 ou 16×16) e as dimensões da imagem o tratamento dado foi: Começar sempre a subdivisão da imagem em blocos a partir da posição $(0, 0)$ e, caso ocorra, desconsideras as possíveis bordas presentes na parte da direita ou na parte de baixo da imagem.

2.4 main_histog_differences(path, stem)

Esse função roda a classe **HistogramDifferences** sobre o conjunto dos valores

$$\alpha = \{3, 4, 5, 6\},$$

que são os valores sugeridos na descrição do trabalho. Nenhuma modificação na forma de se calcular a diferença dos histogramas de quadros consecutivos foi realizada de modo que tal cálculo já está explicitado na parte da secção de introdução que trata desse técnica de identificação de mudanças abruptas entre quadros. No entanto vale destacar que os histogramas foram normalizados de modo que a soma dos valores de seus *bins* totaliza 1.

2.4.1 Detalhes de implementação e execução

Para o cálculo dos histogramas, optou-se por utilizar uma implementação própria ao invés de usar alguma função disponibilizadas pelos módulos do opencv ou do numpy.

2.5 main_edges_differences(path, stem)

Essa função roda a classe **EdgesDifferences** sobre o conjunto de valores

$$MENDS = [0.010, 0.015, 0.020, 0.025, 0.030, 0.035].$$

MENDS é sigla de *Maximum Edges Normalized Difference*. Os valores desse conjunto estão dentro do intervalo $[0, 1]$ e eles representam a diferença máxima normalizada de bordas entre quadros consecutivos tolerada. A normalização é feita considerando a diferença máxima de bordas entre quadros consecutivos possível que ocorre quando um quadro não teria nenhuma borda e o outro quadro teria apenas ‘bordas’. Logo a diferença máxima possível é igual ao número de pixels dos quadros do vídeo analisado.

2.5.1 Detalhes de implementação e execução

Para o cálculo do mapa de bordas, foi utilizado o famoso algoritmo Canny que foi desenvolvido por John F. Canny. Esse algoritmo é bastante popular devido a sua versatilidade e robustez. Ele contempla etapas de redução de ruídos, determinação das intensidades do gradiente da imagem, e aplicação de supressão não máxima e teste de hipóteses. Utilizou-se a versão do algoritmo disponibilizada pelo módulo do opencv *Canny* com os parâmetros de entra fixos. O parâmetro de valor mínimo de 100 e o parâmetro de valor máximo de 200.

Mesmo o algoritmo de detecção de bordas de Canny já conter um etapa de suavização de ruídos, optou-se antes de aplicá-lo, suavizar o quadro de interesse bom um filtro Gaussiano (passa-baixo) de dimensão 5×5 .

3 Resultado

Os resultados levantados são provenientes da aplicação do programa **main.py** sobre as imagens apresentadas na Figura ??.