PCIM 2023 Lab Assignment - Visual Part - assignment 3

Assignment Name: video shot detection
Due Date: May 31st, 2023. Submit in Moodle your report file in PDF and all the Matlab files with the code you have developed as a single compressed archive.

Purpose

To practice the manipulation of video files in Matlab. To understand and experiment with techniques that can segment temporally a video sequence; to understand the role of such process in multimedia applications; to evaluate relevant and alternative techniques.

Why is it important that multimedia students gain the listed skills and knowledge? Because video shot detection is considered to be extremely important in professional video editing and post-production, namely in television and cinema studios. All major editing tools incorporate such functionality to assist the role of the editor.

Skills

In this assignment, you will acquire practice on how to:

1) develop practical algorithms using Matlab to implement video shot detection;
2) manipulate the efficiency of the process, and
3) evaluate the efficiency of the selected techniques.

Knowledge

In this assignment, you'll be learning about:

1) different ways video sequences may be divided into scenes or shots, I.e., the types of transitions that can exist between video scenes/shots;
2) different approaches to temporally segmenting visual signals
3) the parameters that can be manipulated to control the efficiency of the process, and
4) the methodology to adopt to evaluate the efficiency of the process.

Task

In this third work of the visual part, the intention is to detect scene cuts in video sequences, developing two different algorithms in Matlab and comparing their results.

The two different approaches that must be adopted for the development of these algorithms are both based on detecting variations in the video signal. But while one of them does it directly, calculating differences between the values of some characteristic of pixels in consecutive images, the other one does it indirectly, using the motion estimation information.

The first approach involves extracting characteristics of the signal and detecting variations in these characteristics to infer scene-cuts. It works in the uncompressed domain (YUV / YC_bC_r)

The second approach is to experiment with the use of the motion estimation technique. On the one hand, when estimating the backward movement, if it is not possible to find for most of the blocks of the image, candidates (or vectors of movement) that lead to prediction errors below a given threshold, it can be inferred that there is a scene-cut and that therefore a new shot begins in that image. On the other hand, by analyzing the motion vectors generated in a coded sequence, and by comparing their values and orientation between consecutive images, one may also conclude whether or not a scene-cut occurs. Thus, in the latter, it is possible to work in the compressed domain, without having to decompress the sequence of images (generally they arrive at the editing or post-production areas in the compressed format given that most professional cameras already compress the signal).

Generally speaking, the detection of scene cuts in a video sequence corresponds to the identification of sudden variations from one image to the next in a given sequence. To identify automatically these sudden variations that are likely to correspond to scene-cuts, different techniques can be applied, namely:

- variation in some characteristic of the image
    - intensity of the pixels
    - spatial activity or complexity
    - temporal activity of complexity
- number of contours and variation in that number
- comparison of color histograms
- variation in texture + color
- intensity of movement (measurement of the prediction error in the motion estimation technique; intensity and orientation of the motion vectors; reference images used; etc)
- variation in the values of the DCT coefficients

A characteristic of an image is its level of spatial activity. The IAM (Image Activity Measure) provides a measure of the spatial complexity or activity of the image. An expression used to calculate this measure is as follows, in which M represents the number of columns in the image and N the number of rows:

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

Assuming that images of the same scene would have similar IAM values, the occurrence of a difference between these values above a threshold can indicate that there is a cut-off in the scene. Other forms of measuring activity are described ion the accompanying slides.

In any of these techniques, it is necessary to choose a decision level (threshold), above which a given image is considered as a cut of scene. This level of decision can be global and eventually have a value that is adjusted according to experience. It can also have a local/adaptive value. Please see the slides for some thresholding approaches.

A different approach, based not only on the identification and comparison of low-level characteristics of the images, is the one based on the extraction of semantic content existing in the image to decide whether or not different scenes exist in a video sequence. One of the semantic techniques that can be used consists of facial detection and recognition: different people in the images may indicate different scenes.

A final aspect is the assessment of the results. To evaluate the results, you may visualise the original sequence and identify visually the scene cuts. Then you may compare with the results obtained with your algorithms. However, you may also perform some calculations to obtain objective measures. These are most useful when comparing performance between different algorithms. Normally used metrics for assessing accuracy in classification problems include precision and recall, based on the number of occurrences that have been correctly and incorrectly detected (true positives; true negatives; false positives; false negatives). Our problem can also be seen as a classification one: for each new image, the algorithm needs to decide whether to classify it as a new shot or not.

Implementation work to be carried out

In the work to be carried out, it is intended to detect scene cuts using uncompressed video sequences, following the two approaches mentioned above. So <u>two</u> scripts/programs must be developed in Matlab:

1. a script to implement one algorithm of the first approach (detecting variations/discontinuities in one characteristic of the image)
2. a script to implement an algorithm that uses and compares the values of the motion vectors.

1. one (or more) of:

   ii) the measure of the variation of an image characteristic (eg, intensity/luminance);

   iii) contour rate of change

   iv) color histograms;

   v) image Activity Measure or the spatial activity index (see slides)

   vi) temporal activity index (see slides)

The Matlab script to be developed must accept as input parameters, at least a video sequence, and if more than one algorithm is implemented, an indication of the algorithm to use to perform the temporal segmentation. You must return / generate a video file containing only the first images of each identified scene (which are called “keyframes”) and a file with the number of each of these images.

2. For the second approach, a rapid search algorithm is intended to be developed to implement motion estimation (such as 3SS, OSA, or combined directions, as described in classes). To be able to assess the quality of the developed motion estimation algorithm, code available on the course page (Matlab program “EBMA_main”, which calls the functions “EBMA_integer (..)” or “EBMA_half (…)”, can be used to perform the motion estimation with exhaustive search, with full pixel precision or half pixel precision, respectively. This program implements the “exhaustive/full search” algorithm and it is possible to indicate the desired precision. It also uses the function “plot_MV_function (…)”, which allows for displaying the estimated motion vectors. The program accepts two uncompressed images and, using one as a reference estimates the motion vectors for the other using pixel or half-pixel precision. Given that the full-search is the approach to motion estimation that delivers the best result, you can use it to compare the results you obtain with your own fast algorithm.

   As an alternative to the development of a fast search algorithm, students can exclusively use the provided code to carry out the exhaustive search. However, the maximum evaluation that can be obtained in this case in the work developed will be 18 points. Higher scores require the development of a fast search algorithm.

On the course's Moodle page, in addition to a folder with the mentioned Matlab scripts, there are images and video sequences available to be used in the work.

Criteria for Success

It is important that you understand the different approaches commonly used and the factors that control their efficiency as well as the metrics that can be used to assess the efficiency/accuracy of the proposed methods (precision and recall; rate of true positives, false negatives, false positives, etc., vs processing time). Therefore, the report that you should deliver upon completing this assignment should include:

- ✓ a brief description of the video shot detection techniques used, pointing out common aspects and differences and indicating benefits and disadvantages;
- ✓ a brief rationale on how to control the efficiency of each technique and the major effects observed, relating them with the type of content and scene cuts;
- ✓ the methodology and metrics that can be used to evaluate the above-referred efficiency/accuracy;
- ✓ graphics comparing efficiency versus processing time.

The report should be delivered in Moodle by May 31st .

Annex- additional data and information for conducting this assignment

In the scriptsMatlab folder you will find a set of Matlab functions that allow you to work with uncompressed visual information in YUV format. You can use the script 'LoadFileYuv (...)' to import the YUV sequence to Matlab and extract images of that sequence. Proceed as follows:

1. start by viewing the sequence by executing the instructions

    [mov, imgRGB] = loadFileYuv ('filename', 352,288,1: 100); movie (mov)

2. successively save two consecutive (or two images that are separated by a one small number of images), by doing:

    [mov, imgRGB1] = loadFileYuv ('filename', 352,288,10);

    [mov, imgRGB2] = loadFileYuv ('filename', 352,288.11);

Readings

Articles posted on the discipline's website, including the following two: Abdelati Malek Amel, Ahmed Fnaeich, Ben Abdelali Abdessalem, Mtibaa Abdellatif, "Application of motion activity descriptor for shot boundary detection", which defines the measure of image activity; Honghai Yu and Stefan Winkle, "Image Complexity and Spatial Information".

Block of slides "cim2023-videoshotDetection.pdf".

Learn about accuracy metrics in:
https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c.

Note: The template used to describe this assignment is adapted from the University of Las Vegas, Nevada, and the Transparency in Learning and Teaching (TILT) Project.