Lab Assignment 3

Assignment Name: Comparing visual signals

Due Date: April 19th, 2024. Submit in Moodle your report file in PDF and all the Matlab or notebooks/Python scripts with the code you have developed as a single compressed archive.

Purpose

This work intends to apply and consolidate knowledge about the extraction of features from visual signals and use of those features for comparison; to get acquainted with some basic image analysis techniques that can be used to index and query images.

Why is it important that multimedia students gain the listed skills and knowledge? Because it provides them insights on the principles of and practical skills on content based image retrieval, which is an important part of multimedia systems and applications.

Skills

In this assignment, you'll be learning how to:

1) develop simple practical algorithms using Matlab or Python to extract characteristics of still images;
2) use the extracted features to compare images;
3) evaluate the efficiency of the selected techniques.

Knowledge

In this assignment, you'll be learning about:

1) different types of features that can be used to represent visual signals;
2) how histograms can be used regardless of the image spatial dimensions
3) the SIFT algorithm to extract features and generate a feature vector;
4) variations of color histograms to represent images;
5) metrics to establish a comparison between feature vectors.

Task overview

The objective of this work is to allow students to acquire a better understanding of the principles of content based image retrieval, which involves extracting low-level features, generating representative feature vectors and establishing a similarity degree between those vectors.
To carry out these experiments, whenever necessary, use images available on the UC's Moodle as input signals for the algorithms/processes implemented by the developed programs or scripts (some also available on the UC's Moodle page). Many of these images are in PNG (Portable Network Graphics) or BMP (bitmap) format, which means that each pixel is represented by three 8-bit RGB values, that is, a total of 24 bits. By comparing results obtained by applying different extraction techniques, it is intended that the student acquires a better understanding of their efficiency or adequacy to the goal of comparing images.

Example scripts for both Matlab and Python are provided in Moodle in the section devoted to Assignment 3.

Implementation work to be carried out

1. Experiments with color histograms for indexing and querying

As already studied and experimented in the assignment 2, there are several color spaces to represent visual signals. You may re-use your code from the previous assignment to convert between them. Experiments should be carried on using RGB (Red, Green, Blue), HSV (Hue, Saturation, Value, where V represents brightness), and L*a*b (L is the luminosity level, a is color channel along the green-red axis and b is the color channel along the blue-yellow axis. This color space is based in the "color opponent theory", by which no color can be simultaneously green-and-red or blue-and-yellow). Use several images such as "peppers.png", "lighthouse.png" and others available in Matlab, or "floresVermelhas.bmp", "folhasVerdes.bmp", "praia.bmp", available on Moodle. Try with images that you know are similar in color and others that are quite dissimilar.

1.1. Write a script that:

- Imports an image in bitmap format (RGB color space) and displays this image on the screen;
- Is able to convert that image to either the HSV or L*a*b color spaces (the decision to which color space to convert, can be based on the value of a parameter passed to the script);
- For each color space:
  i. Separates the components into separate matrices;
  ii. Generates and plots the color histograms for each of the three channels;
  iii. Concatenates the three histograms into one single vector.

1.2. Write a script that:

- Generates concatenated vectors with color histograms for pairs of images and compares those vectors, evaluating their distance and thus the similarity of images. Whenever it finds that the images are similar it displays them side-by-side. Use different similarity metrics such as Euclidean distance, cosine similarity or intersection distance (see slides "challengesMMstorageRetrieval"). You will need to establish a threshold to take the decision between similar/dissimilar. Run the script with different images and different options for the color space and comment on the results.

1.3. Enhance your script by

- developing a "searchAndRetrieve"" function, which receives one reference/query image and an array of candidate images ("image database"). This function should return the top_k most similar images to the query image, indicated by their index in the array.

1.4. Write a script to

- evaluate the performance of the comparison algorithm by computing the values of Precision, Recall and F1 score (after running the algorithm several times with different images). Based on the obtained results, re-run the algorithm manipulating the resolution of the histogram (number of bins used when generating the histograms) and the value of the threshold. Recall that in a RGB image (bmp or png) each color channel is represented with 8 bits, thus the full resolution histogram will have 256 possible bins (see slides "challengesMMstorageRetrieval"). Explain in your report the impacts that the manipulation of the referred parameters have in the algorithm performance.

3. Experiments with SIFT for indexing and querying

In these experiments, you should investigate the availability of online Python code or existing functions in the Open CV library (cv2 in Python) or in Matlab, that implement the SIFT algorithm. Consequently you should try to use that code to develop a program that allows to generate feature vectors for a pair of images and compute

the distance between those vectors to evaluate similarity between images (the same functionality as requested with the histograms in 1.2).

Criteria for Success

It is important that you familiarise yourself with the generation of different versions of color histograms and with their use for comparing images or searching similar images in a database. Also that there are types of features other than color that can also be used efficiently to compare between images. Upon conducting this assignment you should be able to describe the effects that parameters like the resolution of the histogram and the threshold value have in the performance of the comparison/searching algorithm, supporting such description on the obtained performance metrics. Likewise, to be able to explain the impact of the chosen color space.

Report should be delivered up to the 19th of April in Moodle.

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.