TP2-INF6803
VIDEO PROCESSING AND APPLICATIONS

Jinling Xing 1915481

March 14, 2018

1 Introduction

This TP characterizes two methods used for the description of regions of interest in images, including color histogram and SIFT. I will perform them on the Matlab.

2 Questions

2.1 Question 1

Description: Identify, based on your theoretical understanding of the two methods, which one should be the best of the two in at least THREE specific use cases. For example, which is the best method to use if the size of the compared regions is very large (e.g. 200 pixels by 200 pixels)? Why? And if their content is relatively uniform?

Presentation of the two methods(general principles):

Color histogram is a technique based on color space, such as RGB and HSV. We need to extract features of each channel. Three histograms for each color channel (RGB, HSV, etc.) can be merged into one histogram.

Generally, we need to set several bins of each channel and refill the calculated value to the new matrix of HSV color histogram. Finally, I calculated the cosine similarity of two images of HSV color histogram and got a cosine value to represent the similarity of two images.

SIFT (Scale-invariant feature transform) is a technique to extract the image features by means of convolutions with difference of Gaussian and constructing octaves. On the slides(page 72) of Chapter 4, introducing how to calculate the DOG(Difference of Gaussian).

Based on these theory, I constructed three octaves and calculated the DOG(Difference of Gaussian) both of image 1 and 2. The extracted key-point refilled two new matrices. Finally, I calculated the cosine similarity of two images and got a cosine value to represent the similarity of two images.

Case 1. size of the compared regions is very large (e.g. 200 pixels by 200 pixels)

SIFT is best.

Because of the large pixels of image, I guess we can extract more key-points, at the same time, the color histogram cannot change too much of large pixels.

Case 2. content is relatively uniform

Color histogram is best.

Content is relatively uniform means that the color of these images are similar, based on the HSV color space, the extracted HSV histogram is similar of the uniformed content.

Case 3: the totally different images but including words

SIFT is best.

If use the totally different images but including words, the SIFT will be powerful.

2.2 Question 2

Description: Describe in detail the experiments realized to test the hypotheses of the previous point. Which dataset did you use? What difficulties are there in these? Which evaluation criteria did you use? Did you rely on an external framework to test your code?

Test the hypotheses:
In order to test these three hypotheses, I use the same dataset to test the two methods and then compare the cosine similarity.

Dataset: My dataset I used to compare the similar content of images is IAPR TC-12 , including many kinds of images. I need only two similar images from this dataset.
I also used large images from the apple desktop wallpaper(1920*1080). **Difficult of dataset:**

Generally speaking, the difficult thing I think in this dataset IAPR TC-12 is that If I want to do image retrieval, it is a little bit hard for me. What I have done is to calculate the cosine similarity to evaluate the performance of these two methods.

Evaluation criteria:

I compared the cosine value of these two method based on the following code. I used the cosine value to evaluate the performance.

```
A=sqrt(sum(hsvColorHistogram1.^2));
B=sqrt(sum(hsvColorHistogram2.^2));
C=sum(hsvColorHistogram1.*hsvColorHistogram2);
cos=1-C/(A*B);%cosine value
```

```
A=sqrt(sum(i2.^2));
B=sqrt(sum(i22.^2));
C=sum(i2.*i22);
cos=1-C/(A.*B);%cosine value
```

External framework:

I don't need it.

2.3 Question 3

Description: Describe the implementation of the two studied methods. If you did not write all the code yourself, where does it come from? Did it require modifications? Otherwise, from which papers or websites did you inspire yourself to write it? In all cases, what are the primary parameters of your methods? How did you set their values?

Implementation of color histogram:

- First, I imported all of the images I used to evaluate the performance using a for loop, and took out all the images with 'jpg' as end from the folder, and then took out the images in

order. At this way, we can see on the left side on Matlab, our dataset is imported. If the image size is different, I need to resize the images.

- Second, we transformed the RGB image to HSV image and got the bins of HSV, and the number of all the bins $N = binH * binS * binV$. Find the max of H,S,V.
- I created several matrices and vectors, such as col vector of indexes for later reference and histogram matrix to put value of H,S,V.
- Construct hsv Color Histogram using a for loop.
- Compute cosine similarity of hsv histogram matrices.

Implementation of SIFT:

- First, I imported all of the images using a for loop, and took out all the images with 'jpg' as end from the folder, and then took out the images in order. At this way, we can see on the left side on Matlab, our dataset is imported.
- Second, construct DoGs, including three layers of octave generation.
- Third, Obtaining key point from the image.
- Compute cosine similarity of key-point matrices.

How to write code?

For the color histogram, I looked some examples on MathWork and done it by myself.

For SIFT, This code is built from existing code at MathWork "https://www.mathworks.com/matlabcentral/fileexchange
sift–scale-invariant-feature-transform–algorithm". The computation of DOG and key-points is from the MathWork. What I have done is to use the method on MathWork to extract key-points of two images and compute the cosine similarity of the key-points matrices of these two images.

Primary parameters

For color histogram, the parameters I used is the number of bins.

```
%the bins of H,S,V
binH = 16;
binS = 8;
binV = 8;
N = binH * binS * binV;
```

For SIFT, the parameters I used is layers of octaves. I used three layers of octaves.

2.4 Question 4

Description: Provide the evaluation results from your experiments related to the hypotheses of the first point. Use a proper format for their presentation — tables, figures, ...

For the size of the compared regions is very large (e.g. 1920*1080)
The cosine similarity of color histogram is : cos = 0.9702
The cosine similarity of SIFT is : cos = 0.7849 and it took a very long time to compute the large images.

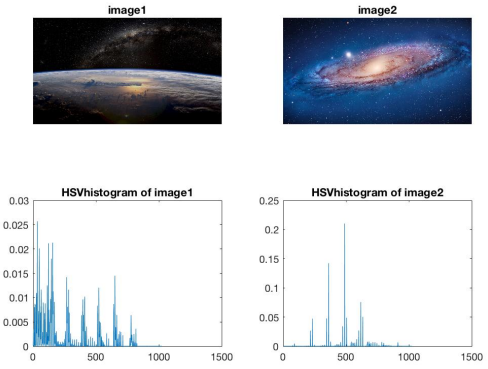

Figure 1: The output of color histogram

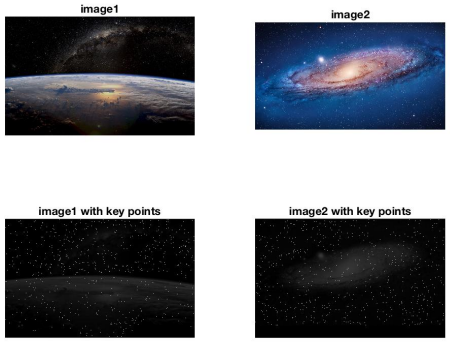

Figure 2: The output of SIFT

For content is relatively uniform:
The cosine similarity of color histogram is : cos = 0.2169
The cosine similarity of SIFT is : cos = 0.2633

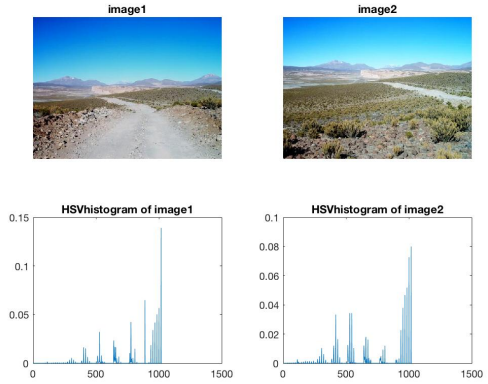

Figure 3: The output of color histogram

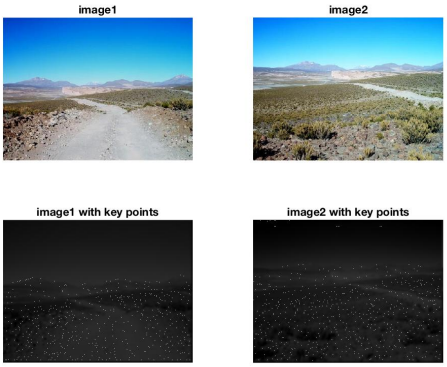

Figure 4: The output of SIFT

The totally different images but including words:
The cosine similarity of color histogram is : cos = 0.8327
The cosine similarity of SIFT is : cos = 0.4183

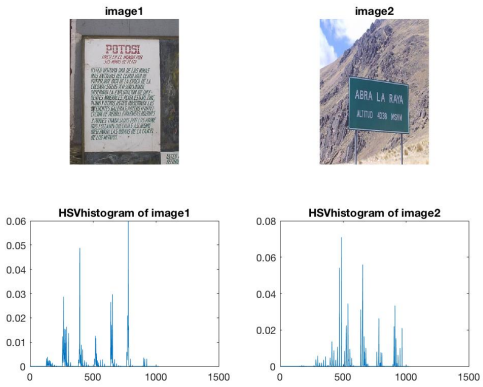

Figure 5: The output of color histogram

Figure 6: The output of SIFT

2.5 Question 5

Description: Discuss the results of the fourth point in relation with the hypotheses of the first point. Which hypotheses are supported by these results? Which are not? Which test resulted in a lack of conclusion? How could you improve these tests?

From the figures above:

Case 1. size of the compared regions is very large (e.g. 200 pixels by 200 pixels)
At the first point: I assumed SIFT is best.
At the fourth point: SIFT is best. The cosine similarity: 0.9702(color histogram) > 0.7849(SIFT)

Case 2. content is relatively uniform
At the first point: I assumed color histogram is best.
At the fourth point: color histogram is best. The cosine similarity: 0.2169(color histogram) < 0.2633(SIFT)

Case 3. the totally different images but including words
At the first point: I assumed SIFT is best.
At the fourth point: SIFT is best. The cosine similarity: 0.8327(color histogram) > 0.4183(SIFT)

Conclusion:

At the case 1: the image is large, which means it has more pixels and the more key-points we can get. Using color histogram, the performance isn't been affected because of the more pixels.

At the case 2: the content is relatively uniform, which means that color space distribution is similar, although the SIFT are powerful and can detected key-points, the color histogram is suited this situation.

At the case 3: the totally different images but including words. What I want to do is that I want to show at the normal situation, which method is best.

How could you improve these tests?

I need to test more images and get more results, using a table to represent the average cosine similarity value of color histogram and SIFT, and the results will be more precise.