CS 1699 Computer Vision

Homework 3

Zihao Zhao

Nov 2015

Part 1. Circle Detection using Hough Transformation

- **No.1**
  As can be seen from the pictures, a low threshold give us more detail image, which means that more lines are presented and there are too many details that the edges may not stand out of the scene. For the high threshold, the image is pretty simple and the edges are obvious. But some edges break up in some areas with high threshold.

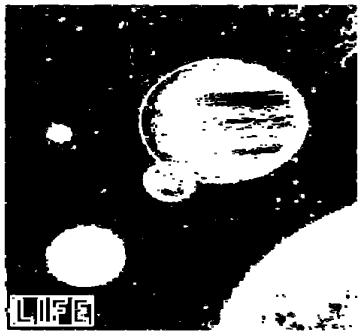


Figure 1: $Threshold = mean * 0.5$

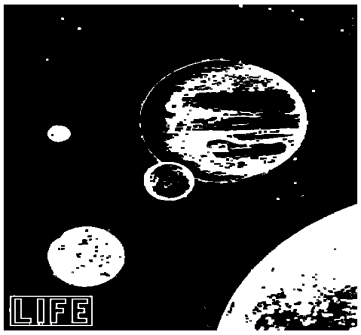


Figure 2: $Threshold = mean * 1.5$

- **No.2**
  If we don't apply gradient orientation, we need to add another loop to loop over all the possible angles($[-\pi, \pi]$) for each pixel point in the image. That also create a new dimension in Accelerator matrix.

- **No.3** For the two picture, I choose 5 radius in randomly in $[1, 20]$, and the results are here.

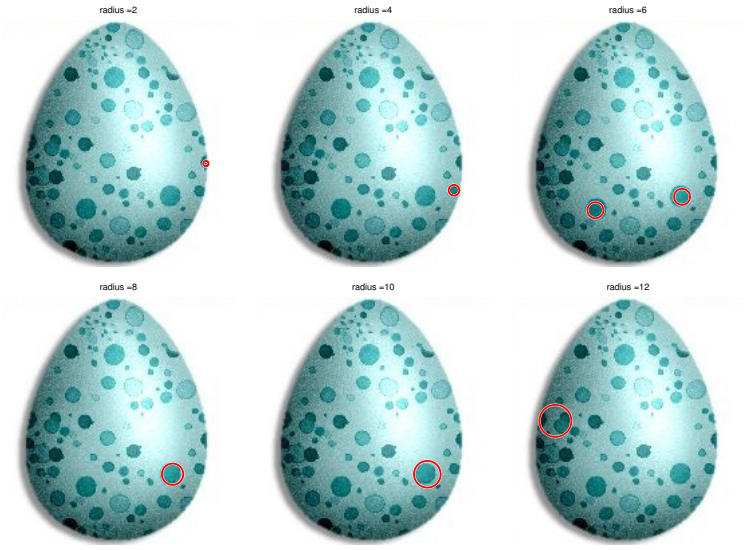


Figure 3: **egg**

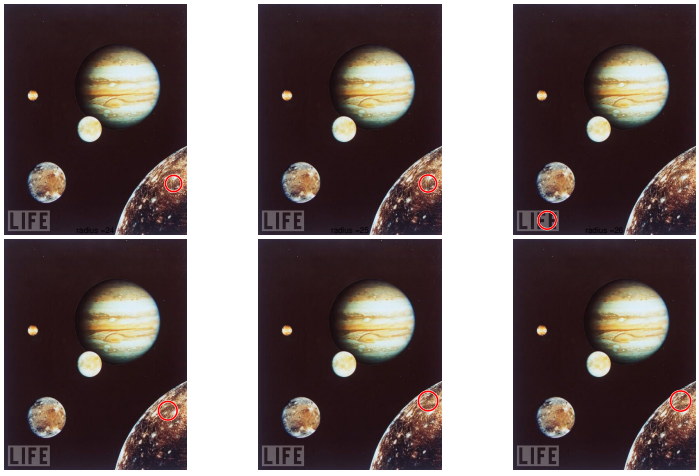
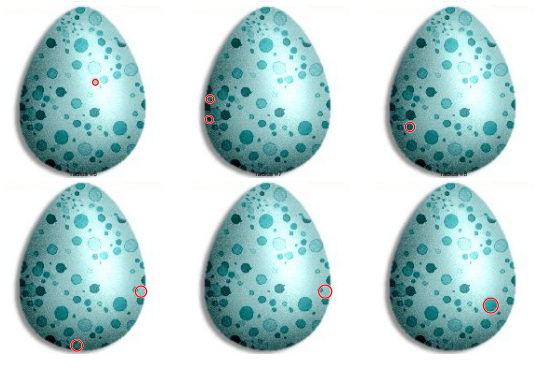


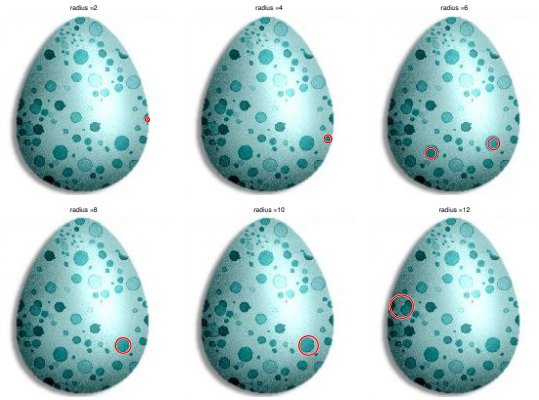
Figure 4: **jupiter**

- **No.4** The different quantization size refers to different results. The larger

the bin size, the smaller size of the accelerator matrix is, the faster to compute the hough space, but at the cost of losing accuracy of the transformation. On the contrary, the smaller the bin size, the larger amount of computation is required, but more precise detection it returns.



(a) Bin szie = 1



(b) Bin szie = 5

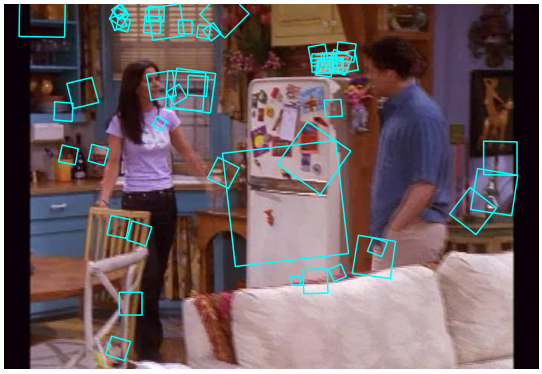
(c) Bin szie = 10

Part 2. Bag of Words

- **No.2 Raw descriptor matching**
  For this part, I select three regions of the one single frame and then match descriptors in that region to descriptors in the second image based on Euclidean distance in SIFT space.

  For the first region, a wooden box is selected and you can see the matched are all over the place. Because I didn't do any sort on the matches I received. However, we still can see that the matches locates around the boxes on the fridge and the brown curtain.

(a) Selected region on frame 1: The Wood Box



(b) All matched descriptors on frame 2.

For the second region, a square of fridge is selected. Again, massive matches are captured along the image, probably because there are too many descriptors in the fridge surface and they got matches with second graph descriptors very well. But still, most of matches locates around the fridge.

(a) Selected region on frame 1: The Fridge

(b) All matched descriptors on frame 2.

This is the third region, I choose the curtain near the window. And for this time the global match receives a great result. All descriptors in second image are around the window.

(a) Selected region on frame 1: The Curtain

(b) All matched descriptors on frame 2.

- **No.3 Visualizing the visual vocabulary**
  The goal of this task was to visualize patches associated with two of the visual words, i.e. we are interested in showing the visual vocabulary. This can be achieved by clustering all feature points of all descriptors of a given data-set. For this purpose I use the assembled mat file that has been generated after the generation of the SIFT data of each extracted video frame. This file contains all SIFT data of every frame in a sequential order that belongs to the same data-set.

  After loading all the SIFT data of every frame, especially all the descriptors (containing all the feature points), I pass that huge set of descriptors to the provided method *kmeansML*. Basically, this method separates all the descriptors (feature points) into k (this is a user specified constant, for this assignment k is equal to 1500) clusters. Each cluster represents a so called word. In the clustering process, each feature points was classified to a particular word. Thus, when selecting two distinct word, make sure their distance is large enough, then we can retrieve the corresponding feature points that belong to that cluster. From this feature point set, we select 25 of them. From those feature points, we select their corresponding patches from the image using the provided Matlab function *getPatchFromSIFTParameters* and visualize all these patches.

Figure 9: **No.1170**

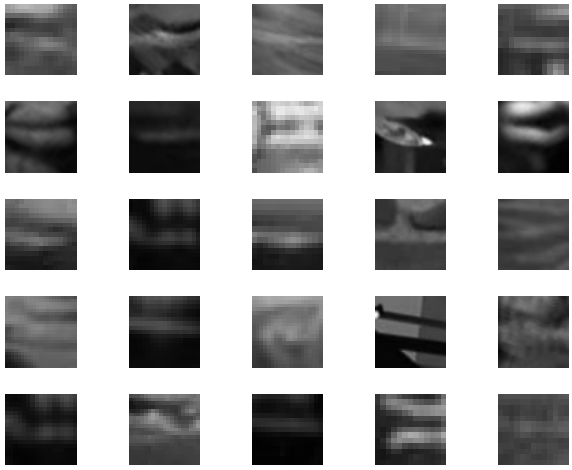


Figure 10: **No.880**

- **No.6 Full frame queries**
  In this section, I choose 3 different frames from entire dataset and use them as queries to search for 5 most similar frames among the frame base. First I use the clustering data from previous computation, and every entry a histogram of clustering. Then I random select 3 different image, extract its name and descriptors, assign them to the nearest visual word to make a bow. After that I use the bow to compare with each data entry in the histogram base and compute the similarity in terms of normalized scalar product. Finally I just sort the similarites and select top 5 frames for the frame base.
  This is the first image, No.70. The query image is the top-left one, and the similar frame are the other. From the figure, we can notice that the actress which serves as the main descriptor of the frame is successfully detected.

Figure 11: **No.7 and 5 similar frames**

This is the second image, No.32. The query image is the top-left one, and the similar frame are the other. The first two frames are great matches for the main characters are detected. For the 3rd, 4th and 6th frames, these frames are detected by the descriptors in the fridges and the table. For the 5th one, it matches with the curtain in the query image.

Figure 12: **No.32 and 5 similar frames**

This is the third image, No.262. The query image is the top-left one, and the similar frame are the other. In this query, the 1st and 3rd frames are great matches for both of the main actors are there. The 5th frame matches for the two actors in first image, but a third actor also comes in the frame and still got matched. The 2nd and 6th are there for the actress is recognized by the query. As for the 4th one, I don't know why it is there to be honest. It's should be the error of detection.

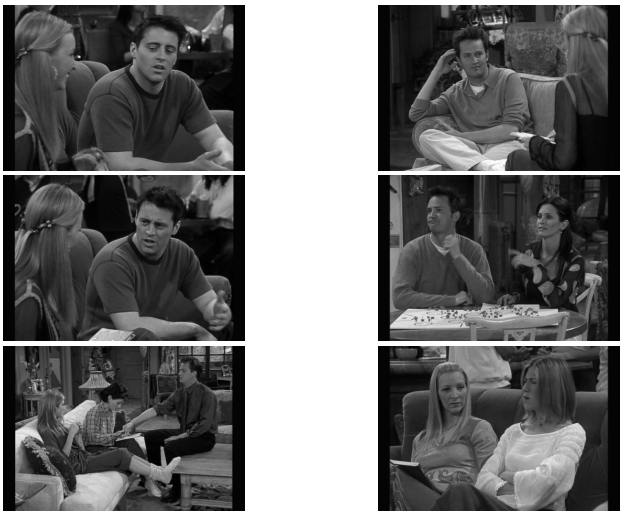
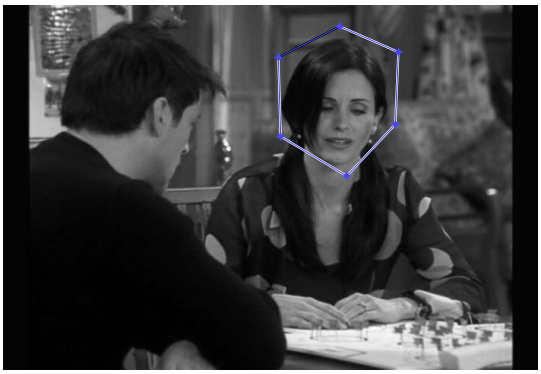


Figure 13: **No.262 and 5 similar frames**

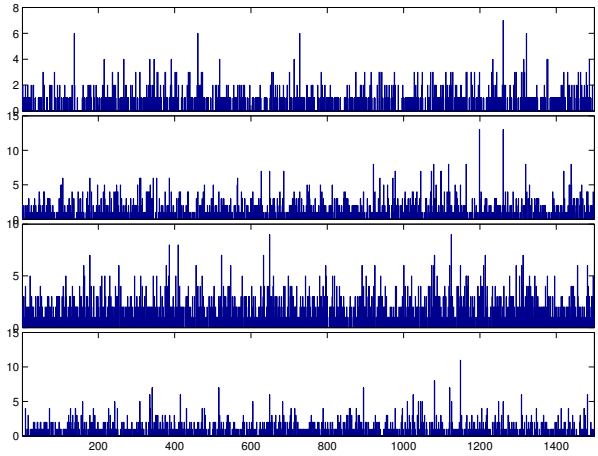
- **No.7 Region Queries**
  This selection is basically same as the previous section, the only difference is that you need to select a region of a frame and use those descriptors fall within the region to match the frames.

  No.80 For this query, I select the actress's face as the query region. The result looks okay since three similar frames all have the actress's appearance. The first two of similar frames match good for the fridge and table, their descriptors are all included in the query image.

Figure 14: **No.80 and 3 similar frames**



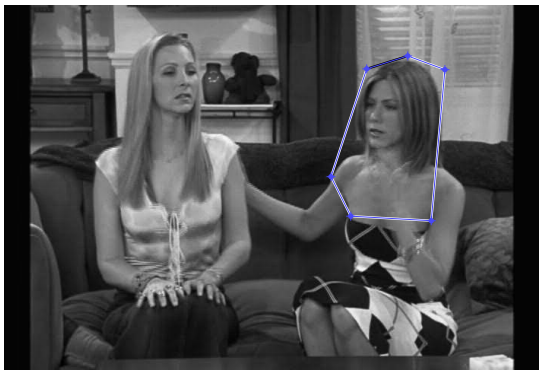
(a) Selected region on No.80



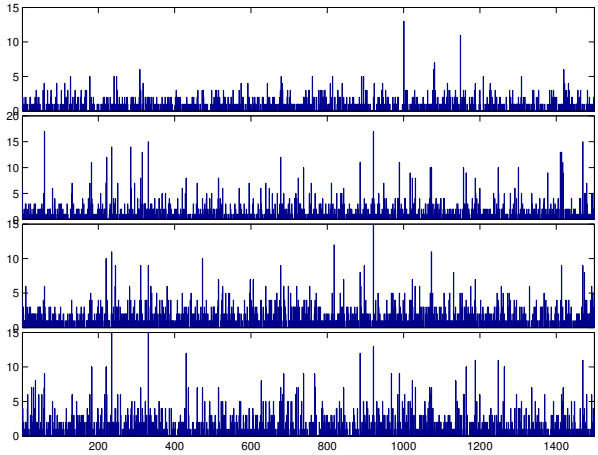
(b) Histogram on 4 frames.

No.270 For this query, I still select the actress's face as the query region, but it's Rachel this time. The result looks great since three similar frames are Rachel's photo. The histogram looks good too. The 3 similar frames's histogram are closely related and part of them have matches from the query histogram.

Figure 16: **No.270 and 3 similar frames**



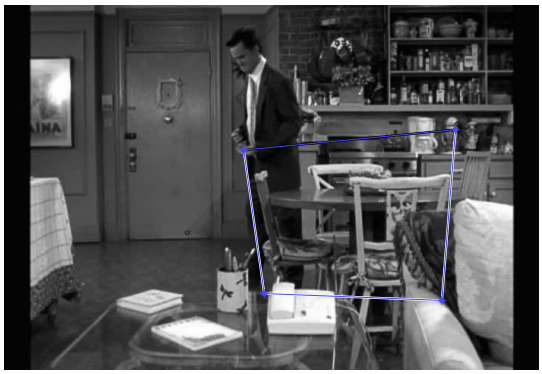
(a) Selected region on No.270



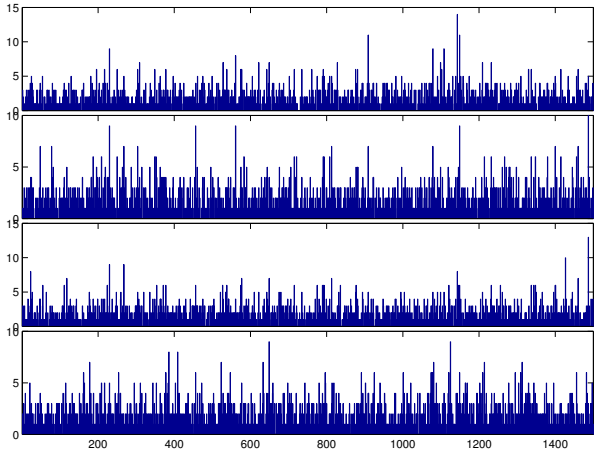
(b) Histogram on 4 frames.

No.128 For this query, I select the desk region to query. The result are almost perfect. All of them are basically the scenes from living room. And the desk stand there in every frames.

Figure 18: **No.128 and 3 similar frames**



(a) Selected region on No.28

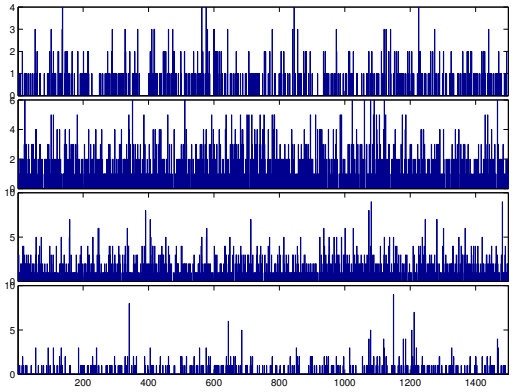


(b) Histogram on 4 frames.

No.28 For this query, I still select the actress's face as the query region, but this time the result is nasty. Only the 1st frames has some matches with query image. The other two are very reluctantly matched. So this one is the failure of region query.

Figure 20: **No.28 and 3 similar frames**

(a) Selected region on No.28



(b) Histogram on 4 frames.