CSCI – 576 – Multimedia Systems Design
Spring 2014 Project

Media Based Querying and Searching

Project Members:

Manan Vijay Vyas
USCID: 7483-8632-00
Email: mvyas@usc.edu

Shreyansh Dnyanesh Daga
USCID: 6375-3348-33
Email: sdaga@usc.edu

Algortihms Used:

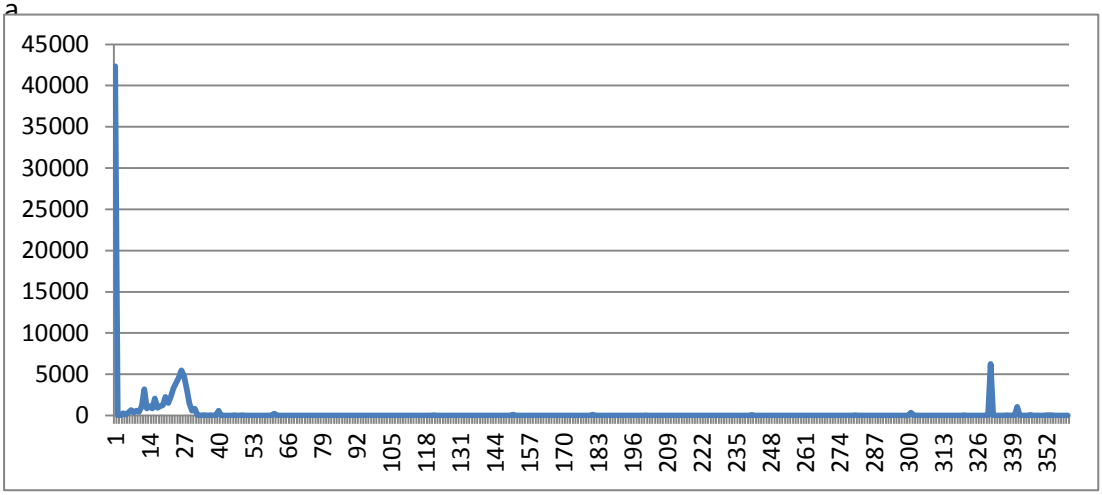
1. Color Space Conversion from RGB to HSV space and analysing its Histogram for Query and Database Images.

To find out the dominant hue within the query image, RGB->HSV conversion was performed.

Then a histogram of the HSV space was calculated with an optimum bin size for the H and S space for the query image.

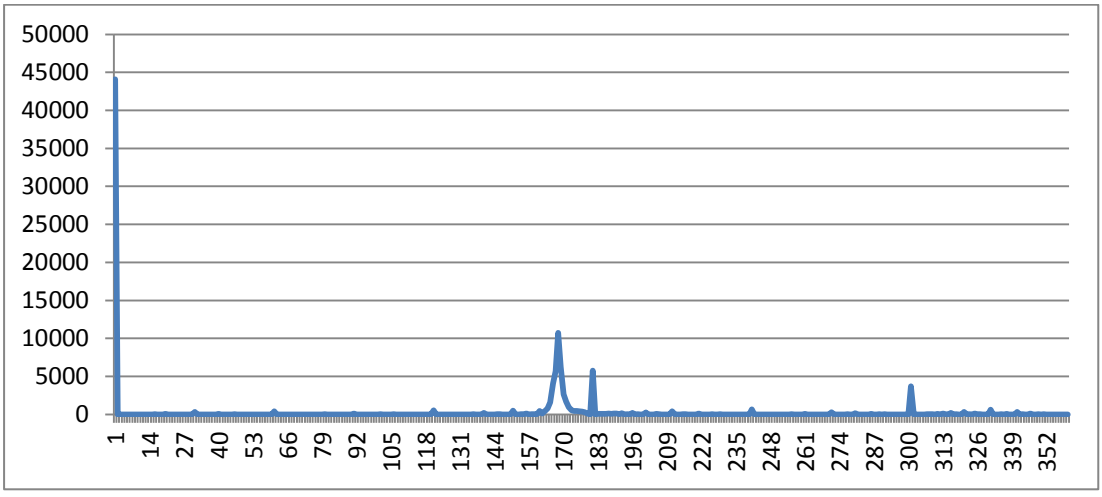
This gave us a rough idea of what is the 'major' color in the Query image and its s

a



Plot[1] : Histogram of Hue of Coca-Cola Source image.

Histogram of Orange Source image dominant component around 27deg. With orange Hue



Plot[2] : Histogram of Hue of Starbucks Source image.

And this was we determined thee histograms for the source images as well as the sub-blocks of database images.

Now, in order to find the best match, we divided the histogram into different bins. Eg. 180 bins of a Hue histogram will correspond to 2deg in every bin. But just using Hue mistogram matching was not enough as there is no guarantee that only pure coloured components of the query image will be present in the database image. Eg: Pure red and Dull red have the same Hue but different saturation. Hence, we decided to use Hue and Saturation histograms for matching purposes.

Comparing the Histogram bins of the Query Image and that of areas within the different areas and comparing the mean squared error between the two and returning the blocks that have the least mean squared error.

The choice of blocksize is also crucial, and hence we used different methods to obtain the optimum result. Divide and Conquer was also used, but better results were obtained when we kept a fixed window size and mark out the candidate blocks based on MSE threshold and then combining them to form a large block that contained the image that we are looking for.

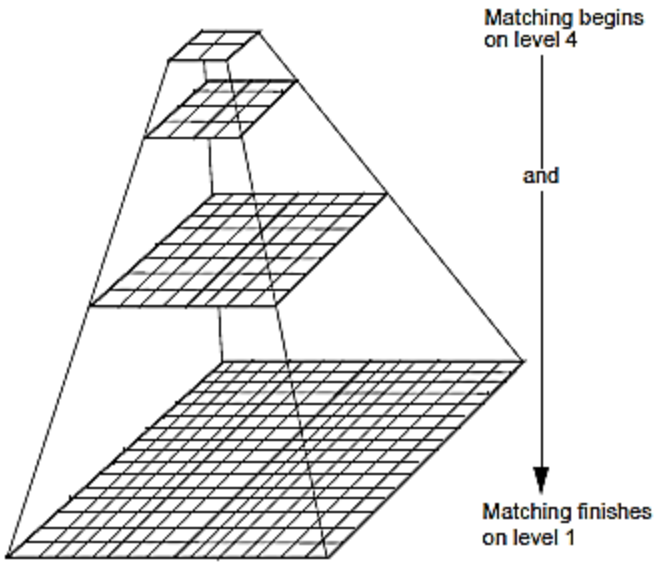
2. Template Matching with Pyramidal Resampling of Query Images.

We also used the Template Matching algorithm from OpenCV with minimum correlation for this project with the histogram matching(Hue+Saturation) metric of mimimum CV_BHATTACHARYA (Bhattacharya distance) giving the best results of all the possible distance metric.

For Template Matching, of the main things to be taken care of is that, the template should be a subset of the image within which the template has to be sought. Hence we used a pyramidal re-sampling technique to generate the template from the Query images and then apply the OpenCV template matching algorithm in our project.

Various matching parameters were used for matching criteria like squared difference and correlation (correlation gave best results).

The pyramidal resampling can be explained by the following image;



Fig[1] : Pyramidal Subsampling scheme used to take into consideration scaling effects.

Here we used 3 levels of Pyramidal Subsampling.

Reason for applying Pyramidal Subsampling: The search images may have a scaled down image of the query image with different orientations. Hence we progressively decrease the block size and traverse around the image to find a matching area that is closest to the HSV histogram of the query image.

Pros: Best Results obtained when Template Matching with Pyramidal Subsampling was used alongwith the color histogram analysis and segmentation described in 1.

Cons: The template has to a subset of the database image and the template matching always gives a best match, and hence the danger of getting a false positive is also present.

3. Color Segmentation and Clustering using K-Means and Mean Shift Filtering :

The idea behind using any clustering or segmentation algorithm was to enhance the areas containing dominant hues so that they will be earlier to recognize in subsequent steps. Hence, we used the two most common clustering and segmentation techniques which are as follows.

- **Mean Shift Filtering:**

Mean shift is a nonparametric iterative algorithm or a nonparametric density gradient estimation using a generalized kernel approach.

1. For each point xi
2. Choose a search window
3. Compute the mean shift vector m(xi,t)

4. Repeat till convergence.

Let {xi} i=1…n be the original image points, {zi} i=1… n be the points of convergence, and {Li} i=1…n a set of labels.

1. For each i = 1…n run the mean shift procedure for xi and store the convergence point in zi.
2. Identify clusters {Cp} p=1…m of convergence points by linking together all zi which are closer than 0,5 from each other in the joint domain.
3. For each i = 1…n assign Li = {p | zi∈ Cp}.

Pros of Mean Shift Filtering:

- Well Defined Color Segmentation, gives excellent segmented hues for further processing

Cons of Mean Shift Filtering:

- Takes a very long time to run, a good optimized implementation required almost 15 mins to run and hence this method, even though gave good results, it was discarded.
- Time Complexity very high : $O(T.n^2)$ where 'T' is number of convergent iterations till stopping condition of minimum mean between consecutive clusters is met and 'n' is the total number of pixels in the image.

- **KMeans Clustering:**

It was also used to segment colors but its biggest negative point being, it clusters a variety of colours that are nearby themselves in a single cluster and hence rich hue information was lost.

Pros of K-Means:

- Fast with Time Complexity of O(k.n.T) where ; 'k' is the number of intended clusters, 'n' is the number of pixels in the image and 'T' is the number of convergent iterations.

Cons of K-Means:

- K-means is very sensitive to initializations. A wrong initialization can delay convergence or sometimes even result in wrong clusters.
- K-means is sensitive to noisy outliers.

The best results were obtained when a combination of histogram analysis and Hue based searching with block based searching using Pyramidal Resampling using the minimum Bhattacharya distance alongwith template matching in-built of OpenCV with maximum Correlation as its matching criteria.

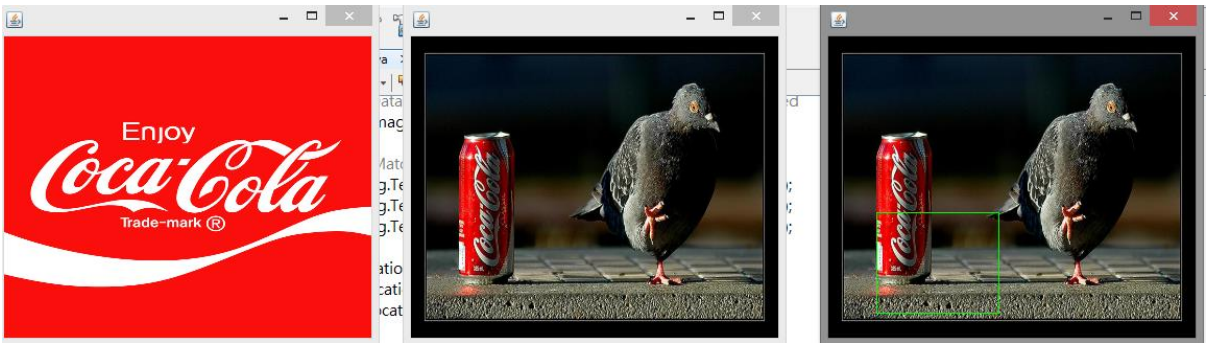
Results:

Here the Extreme LHS image is the Alpha image/Original Source image. The middle image is the image on which the LHS was to be found and Extreme right image contains the output of our approach with a green coloured bounding box denoting the area that is detected that 'contains' the query image.

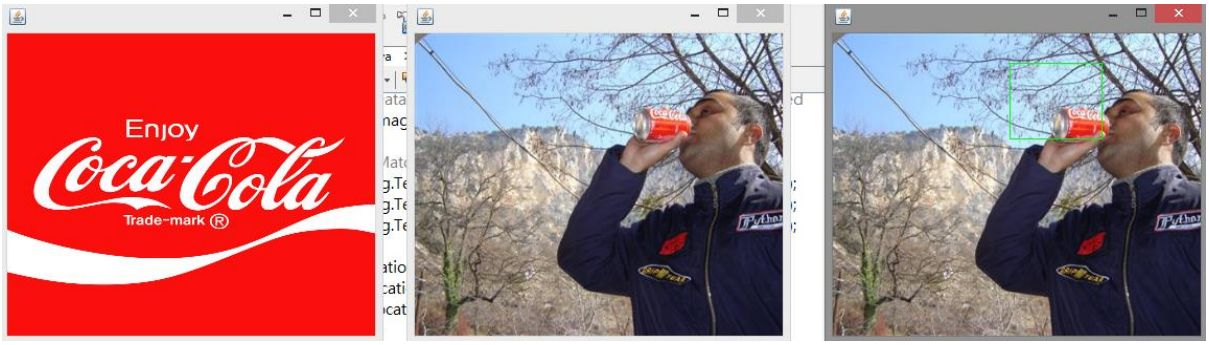
Fig[2] : Result of USA Flag Level 0.

Fig[3] : Result of USA Flag Level 1.

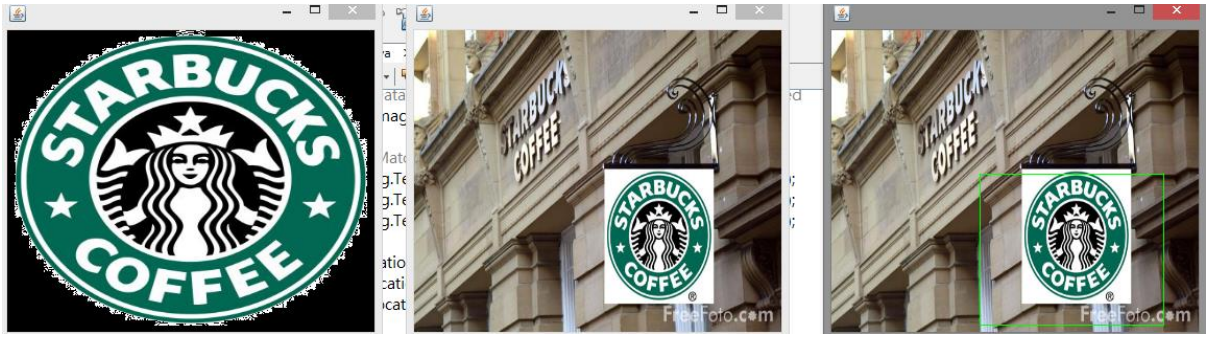
Fig[4] : Result of USA Flag Level 2.



Fig[5] : Result of Coca-Cola Level 1.

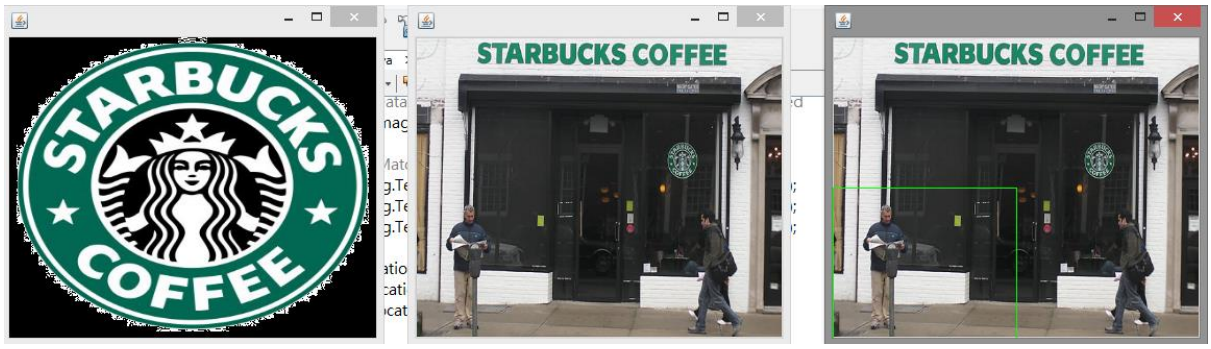


Fig[6] : Result of Coca-Cola Level 2.

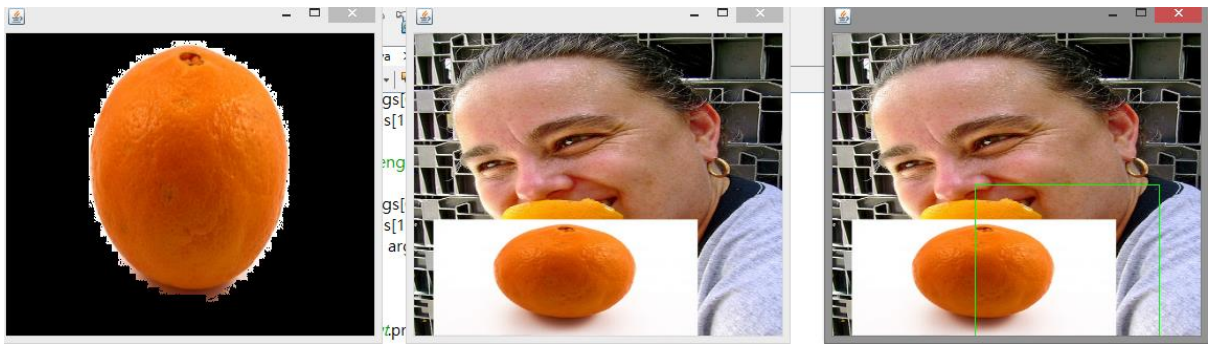


Fig[7] : Result of Starbucks Level 0.

Fig[8] : Result of Starbucks Level 1.



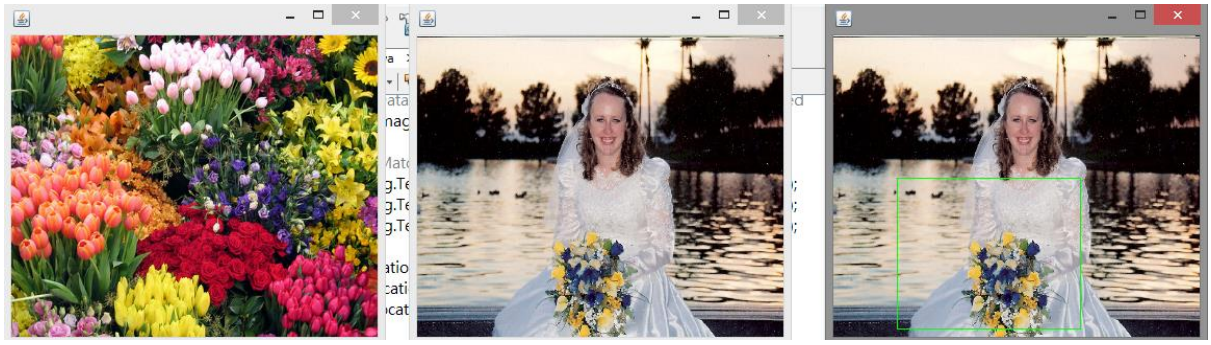
Fig[9] : Result of Starbucks Level 2. (Gives False Positive)



Fig[10] : Result of Orange Level 1.

Fig[11] : Result of Orange Level 1.

Fig[12] : Result of Orange Level 2. (Gives False Positive).



Fig[13] : Result of Flowers Level 1.

Fig[14] : Result of Orange Level 2.