# MCA ASSIGNMENT 1 REPORT

## Color Auto Correlogram

To implement the Color Auto Correlogram

- The input images were first resized to 300x300.
- The 256*256*256 RGB colors were quantized to 4*4*4, which gave us 64 color bins.
- Color Auto Correlograms obtained for all images.

The algorithm for the Color Auto Correlogram was made by referring to the paper: http://www.cs.cornell.edu/~rdz/Papers/Huang-CVPR97.pdf.

The correlogram considers all of the neighbors of a pixel which are d distance away.
The distances considered were: (1, 3, 5, 7, 9).
To speed up the construction of the algorithm, every d/2th pixel was checked for a particular distance d, and every neighbor of the given pixel was considered.

For image retrieval, the following ranking system was used:

$$|\mathcal{I} - \mathcal{I}'|_{\gamma,d_1} \triangleq \sum_{i,j\in[m],k\in[d]} \frac{|\gamma^{(k)}_{c_i,c_j}(\mathcal{I}) - \gamma^{(k)}_{c_i,c_j}(\mathcal{I}')|}{1 + \gamma^{(k)}_{c_i,c_j}(\mathcal{I}) + \gamma^{(k)}_{c_i,c_j}(\mathcal{I}')}$$

Here I and I' refer to the two images, ɣ being their correlograms, setting ci=cj, we obtain values for our correlogram. 'k' here refers to the distance d.

The **top 100** images from the retrieval system were considered.

The following evaluation metrics were used:

- Precision: The number of relevant images retrieved by the total number of images retrieved (in our case, 100).
- Recall: The number of relevant images retrieved by the total number of relevant images (the total number of images in the ground truth files good, ok and junk for a given query).
- F1- Score: The harmonic mean of precision and recall.

The results of the retrieval system are given below:

Manan Gupta
2017372

| Query | Precision(%) | Recall(%) | F1-Score | Good | Ok | Junk | Time (s) |
|---|---|---|---|---|---|---|---|
| all_souls_1 | 4 | 3.6 | 0.04 | 1 | 3 | 0 | 5.4 |
| all_souls_2 | 3 | 2.7 | 0.03 | 1 | 2 | 0 | 5.4 |
| all_souls_3 | 10 | 9.01 | 0.09 | 5 | 3 | 2 | 5.26 |
| ashmolean_1 | 6 | 19.35 | 0.09 | 5 | 1 | 0 | 5.6 |
| ashmolean_2 | 4 | 12.9 | 0.06 | 4 | 0 | 0 | 5.72 |
| ashmolean_3 | 2 | 6.45 | 0.03 | 0 | 2 | 0 | 5.87 |
| balliol_1 | 5 | **27.78** | 0.08 | 2 | 2 | 1 | 5.78 |
| balliol_2 | 3 | 16.67 | 0.05 | 0 | 2 | 1 | 5.35 |
| balliol_3 | 2 | 11.11 | 0.03 | 0 | 2 | 0 | 5.34 |
| bodleian_1 | 4 | 13.33 | 0.06 | 2 | 2 | 0 | 5.36 |
| bodleian_2 | 4 | 13.33 | 0.06 | 3 | 1 | 0 | 5.37 |
| bodleian_3 | 3 | 10 | 0.05 | 2 | 1 | 0 | 5.38 |
| christ_church_1 | 11 | 8.27 | 0.09 | 8 | 0 | 3 | 5.51 |
| christ_church_2 | 12 | 9.02 | 0.1 | 7 | 1 | 4 | 5.51 |
| christ_church_3 | 10 | 7.52 | 0.09 | 6 | 2 | 2 | 5.82 |
| cornmarket_1 | 2 | 15.38 | 0.04 | 1 | 1 | 0 | 6.85 |
| cornmarket_2 | 2 | 15.38 | 0.04 | 0 | 2 | 0 | 5.3 |
| cornmarket_3 | 2 | 15.38 | 0.04 | 2 | 0 | 0 | 5.27 |
| hertford_1 | 5 | 8.2 | 0.06 | 4 | 1 | 0 | 5.31 |
| hertford_2 | 14 | 22.95 | **0.17** | 11 | 3 | 0 | 6.2 |
| hertford_3 | 4 | 6.56 | 0.05 | 2 | 2 | 0 | 5.85 |
| keble_1 | 1 | 9.09 | 0.02 | 1 | 0 | 0 | 5.3 |
| keble_2 | 1 | 9.09 | 0.02 | 1 | 0 | 0 | 5.16 |
| keble_3 | 1 | 9.09 | 0.02 | 1 | 0 | 0 | 5.03 |
| magdalen_1 | 5 | 4.85 | 0.05 | 3 | 0 | 2 | 5.13 |
| magdalen_2 | 7 | 6.8 | 0.07 | 1 | 4 | 2 | 5.25 |
| magdalen_3 | 2 | 1.94 | 0.02 | 1 | 1 | 0 | 5.52 |
| pitt_rivers_1 | 1 | 12.5 | 0.02 | 1 | 0 | 0 | 5.07 |
| pitt_rivers_2 | 1 | 12.5 | 0.02 | 1 | 0 | 0 | 5.07 |
| pitt_rivers_3 | 1 | 12.5 | 0.02 | 1 | 0 | 0 | 5.16 |
| radcliffe_camera_1 | **30** | 8.62 | 0.13 | **16** | 9 | **5** | 5.13 |
| radcliffe_camera_2 | 27 | 7.76 | 0.12 | 12 | **10** | **5** | 5.16 |
| radcliffe_camera_3 | 15 | 4.31 | 0.07 | 5 | 7 | 3 | 5.37 |

Manan Gupta
2017372

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Mean | 6.18 | 10.72 | 0.058 | 3.33 | 1.93 | 0.90 | 5.44 |
| Max | 30 | 27.78 | 0.17 | 16 | 10 | 5 | 6.85 |
| Minimum | 1 | 1.94 | 0.02 | 0 | 0 | 0 | 5.03 |

The mean, max, and minimum of all the queries are given above.

### Analysis:

Largely speaking, the results obtained for the image retrieval using the color autocorrelogram were fairly poor. This can be because of the loss of information caused by certain steps taken to reduce autocorrelogram construction time such as the **small sizes of the resized images**, as well as the **skipping between pixels**. This allowed for rapid prototyping and the entire algorithm ran on the 5000+ images under 6 hours. Some tests were performed, if the algorithm was run with the computational resources my PC has, without the short-cuts taken, it would take upwards of 2 days to execute.

## Scale Invariant Feature Transformation (SIFT)

The algorithm was referenced from the site.
The SIFT algorithm has 3 main steps:

1. Generation of LOG filters:
    a. The LoG function takes sigma as input and returns a filter with size 6*sigma.
2. Convolving the images with Gaussian filters
    a. The images are convulsed over with an LoG filter with different sigma values. The different sigma filter allows finding features of different sizes, thus scale invariance.
3. Finding the maximum peak
    a. We have considered 3x3 neighborhoods. Each neighborhood will have a maximum peak. But not all neighborhoods would contribute to a detected blob. So we can work around with the threshold to find the best peaks.
    b. Once these peaks are located, the center is reported, along with the corresponding radius.
4. One may additionally choose to remove all redundant blobs, since many of them may coincide.

Files:

- File1.py: Code used to construct color auto correlogram
- File2.py: Code used to obtain features using SIFT
- ac.p: Pickle file containing the dictionary object with Color Autocorrelograms for all images.
- sift.p: Pickle file containing the dictionary object with coordinates of all blobs using SIFT algorithm for all images.
- autocorrelogramQuery.py: File used to query images using the correlograms obtained from File1.py.

Manan Gupta
2017372