Object Recognition Exercise

Henrik Aanæs Anders Dahl

April 17, 2013

The aim of this exercise is to give you a feel for how image database search can be carried out, and often is. In order for the workload of this exercise to be reasonable, you will have to make use of several code packages from the internet.

An outline of this exercise is, that you are given 100 images, of these you should use 99 to make an 'image data base', i.e.

1. Extract SIFT features from all the images.
2. Find K clusters of these sift features (there are typically 100.000 to 200.000 features in these 100 images), via k-means clustering.
3. Based on these K clusters, a K-bin histogram is formed for each image, denoting how many extracted sift features are closest to a given cluster. This is seen as a descriptor for the image.
4. Calculate the image descriptor for the image left out, and find the images it's descriptor are closest to.

1 Extracting Sift Features

Unzip the images located in `ukbench.zip`, these images are provided by Henrik Stewenius et al. from the university of Kentucky, see `http://www.vis.uky.edu/~stewe/ukbench/`. This data set has been extended, and the small original version is located together with this exercise.

For calculating the SIFT features you should download the code from the inventors homepage, i.e. `http://www.cs.ubc.ca/~lowe/keypoints/` or the `vl_feat` framework. A sample script for extracting the features via Lowe's code is:

```
Images=[];
for cIm=0:(nIm-1),
if(cIm<10)
name=sprintf('ukbench0000%d.jpg',cIm);
elseif(cIm<100)
name=sprintf('ukbench000%d.jpg',cIm);
end
Images{cIm+1}=rgb2gray(imread(name));
```

```
end

SIFTdescr=[];
for cIm=1:nIm,
imwrite(Images{cIm},'HaaScript.pgm');
[image, descrips, locs] = sift('HaaScript.pgm');
SIFTdescr{cIm}=descrips;
end
```

First try with 20 images and 50 clusters, to get your code running, then try with all 100. It is noted that 50 clusters is very few, and most decent size systems will have considerably more. It is recommended that you use so few here for speed for computing.

2 K-Means Clustering

MatLab has a built in k-means clustering algorithm, but it does not seem to perform fast enough for this purpose. Instead we recommend that you download and use the K-means clustering matlab code from
`http://cseweb.ucsd.edu/~ elkan/fastkmeans.html`. Find the K(=50) clusters.

remember not to include the query image in the clustering!

3 Constructing Image Descriptors

For each SIFT feature in each image determine which cluster it is closest to. Then for each image make a K-bin histogram of how many SIFT features are closest to each of the K clusters.

Lastly compare the distance between the hisogram of the image to be compared to the rest of the images. This should be done via a χ^2 distance, ie

$$dist(\mathbf{x},\mathbf{y}) = \sum_i \frac{(x_i - y_i)^2}{x_i + y_i}$$

where the term is set to zero if $x_i + y_i = 0$.

How well does it perform?