MVA 2008 - 2009

Object recognition and computer vision

Assignment 1: Scale-invariant blob detection

Antoine Labatie

I implemented the blob detection with difference of gaussians to approximate the Laplacian and by dividing the scale-space in octaves and downsampling as in David Lowe's paper.

Implementation choices

Speed

I have realised Matlab is very long to do loops so I tried to avoid it.

I used filtcol with a function acting on every column of the matrix created by filtcol (every column represents a block) to detect maximums in each image.

To find a maximum in the scale-space, I use the first matrix created with filtcol to know if their is a maximum in the image. If their is one, I compare the maximum in the [3 3] neighbourhood at this level with the maximums in the [3 3] neighbourhoods of adjacent scales.

I get matrices for each octave j and level i, which say if a point (k,l) is a local maximum in scale space. I created a vector for the blobs, which is (Nb_Blobs,3) where the 3 components correspond to the position and the sigma. I had to do a « for » loop to insert them by exploring the matrices at octave j, level i, but I started with a test to skip when there are no maximums at one level or in one row.

Potential differences with SIFT

I was not sure how those different issues are dealt with in the real SIFT method so I probably made something a little different.

Before downsampling I started from the normal image but smoothed it enough so that there is not much information lost (by a factor $0.6*2^j$ for j^{th} downsampling).

I chose an increment « k » equals to $2^{1/N}$ as in Lowe's paper, and computed N+2 Laplacians so that I can detect maximums from the level 2 to the level N+1.

I also worked with « blurred » images, that is O starting with O=2 relatively to the image size in the new octave.

Parameter values

The different parameters are:
-O the number of octaves
-N the number of levels in each octave
-« k » the factor by which sigma is multiplied each time
-the threshold on $(k-1)^2*O^2/\infty^2$ for blob detection

-For the number of octaves I chose 3 and when I tried more I usually found not relevant regions.

-I found a good value for N is 5, since it detects enough extremum, and when I tried with more levels, there are lots of repetition in the blobs detected (what seems to be a single blob is detected several times).

-Concerning « k » I finally set it at $2^{1/N}$ as in Lowe's paper. I tried before to set it as $3^{1/N}$ or $4^{1/N}$ which seemed sometimes to perform better (at least to detect more blobs). Yet I realised some blobs are detected several times because there are overlappings between octaves in terms of sigma, and so the same maximum may be detected in different octaves.

-I found the threshold is the hardest issue and could not really find a good value since it seems to depend on the image. For some images as the « fishes » of the « sunflowers », the threshold needs to be much lower than for the « butterfly » in order to detect the main blobs. So the threshold should be somewhere around 60 and 300. As a consequence I finally used a threshold relative to the image: I compute the maximum of the Laplacian values and keep points with a Laplacian> 0.4*GlobalMaxLaplacian.
I was also a bit puzzled about the threshold value, since it does not correspond at all to the value you suggest (that is 100 instead of 0.001). I could not see where I made mistake if I did, since I just made the difference of the Gaussians which immediately gives the scale invariant value (k-1)*O/ ∞ from which I take the square value.

Results