TP2: Active Shape Models (2)

Loic Simon Chaohui Wang Panagiotis Koutsourakis

14 Mai 2007

1 Theory

1.1 Principal Component analysis (PCA)

PCA is a common technique used to express the variation of a given data set around its mean and along some well chosen eigenmodes. Practically it involves the three steps below:

- We compute the mean of the data and center them
- We calculate the covariance matrix (which is real symmetric and so can be diagonalizable in an orthonormal basis)
- Eventually, we keep only the k most relevant eigenvectors in the sens that are associated to the highest eigenvalues and therefore they concentrate as much variance as possible from the data.

One can consider in Figure 1 a simple PCA result when applied to a set of 2D points. The 2 modes give the main axes of an ellipse.

1.2 Application to Active Shape Model (ASM)

A contour can be represented as a set of m pairwise connected 2D points which coordinate will be denoted by $(x_i, y_i)_{i=1\cdots m}$. Then we can regard a contour as a vector $\mathbf{X} = [x_1\ y_1 \cdots x_m\ y_m]^t$, whose dimensionality is $2m$.

Let's assume that we know n contour $\mathbf{X_i}$ for $i = 1 \cdots n$. Once the mean contour has been computed using:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} \mathbf{X_i} \tag{1}$$

we may obtain the covariance matrix as follows:

$$\mathbf{C} = \frac{1}{n}\sum_{i=1}^{n} (\mathbf{X_i} - \mu)(\mathbf{X_i} - \mu)^t \tag{2}$$

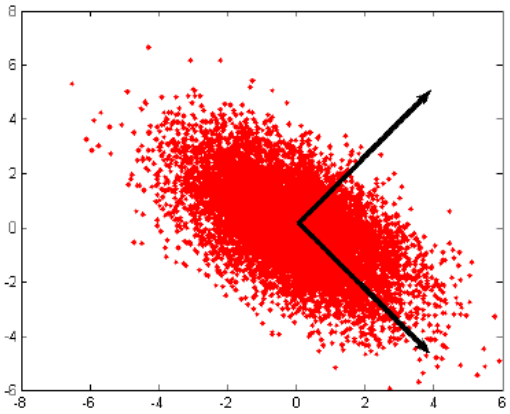

Figure 1: PCA over a set of 2D points normally distributed

Then analyzing a new contour $\mathbf{X_{New}}$ consists in projecting it on the affine subspace passing by the mean contour and spanned by the k major eigenvectors:

$$\mathbf{X}_{\mathbf{New}}^{\mathbf{Ana}} = \mu + \sum_{i=1}^{k} \alpha_i e_i \tag{3}$$

For Gaussian samples, it's commonly accepted that a new contour has a high likelihood if its components along the basis validate $|\alpha_i| \leq 3\sqrt{\lambda_i}$.

2 Computation : PCA on hand contours

The main goal of this lab is to implement PCA and discuss the results obtained with hand contour. In particular, it will be interesting to observe the variation of the mean contour along the eigenmodes.

1. Create a project in your IDE and add the contour.h/cpp, data.h/cpp and pca_main.cpp files in it.

   Note.: In contour.h/cpp, a Contour class is defined which encodes a contour. It has a constructor function which can create an object by loading the data from a file describing the contour. And in data.h/cpp a Data class is defined, which can carry out the PCA to a set of data. It has a function which translates a set of contour into a set of data.

2. Complete the function

```
void Data::mean_covariance(CImg<double> & mean, CImg<
    double> & covariance);
```

in the file data.cpp

3. Launch the program and play with the sliders to make vary the contour around the average contour and along the first principal components.

4. The sliders allow to vary between 0 and $3 * sqrt(\lambda_i)$. Do you think all contour obtained are realistic? What can you conclude from this?

3 Snakes computation exercises

This exercise is for those who didn't finish the previous lab. Here the code is complete. You just have to modify some global variables and comment the results.

3.1 Internal versus External energy

Test the algorithm with different values for the respective weight of the internal and external energies (to do so play with the three global variables wElas, wStiff and wExt defined at the top of snake.cpp). In particular try it respectively without one of the two components of the energy (wElast=wStiff=0 then wExt=0). What are the problems in each case?

3.2 Bluring the input image

Evaluate the behavior of the algorithm without bluring the input image (put sigma to 0 in snake.cpp). For this question, use rectangle.png and rectangleNoisy.png. What can you notice (benefits and drawbacks of the blur)?

3.3 Bonus question

What are the principal drawbacks of the current implementation. What can you propose to improve it.