

# Computer Vision Project [H02K5a]

Incisor Segmentation

Glenn Cools, r0370846  
Hendrik Van Hove, r0341925

KULeuven, Belgium

26th June 2017

## 1 Introduction

Biomedical images usually contain complex objects, which will vary in appearance significantly from one image to another. Attempting to measure or detect the presence of particular structures in such images can be a daunting task. The inherent variability will thwart naive schemes. However, by using models which can cope with the variability it is possible to successfully analyse complex images [1]. In this way, it is possible to examine dental radiographs to indicate positions of incisors automatically using a clever computer program. This can be very helpful in for example forensic investigations, where identifying people post-mortem can prove to be a very tedious task when done manually.

This report discusses our solution for the Incisor Segmentation project of the course Computer Vision [H02K5a], with as goal developing a model-based segmentation approach, capable of segmenting upper and lower incisors in panoramic radiographs. We will start the paper by explaining the construction of an Active Shape Model in section 2. In section 3, we discuss the different pre-processing steps applied to the photographs to facilitate the fitting of our model. Next, we attempt to fit the generated models on an image in section 4. This part consists of finding an initial estimate for the incisor central position followed by iteratively improving the positions of the outline of the incisor. Finally, in section 5 and 6 we discuss the evaluation of the model and reflect back on the project.

## 2 Active Shape Model

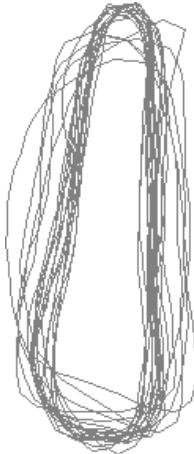
Active Shape Models are statistical models of shape of objects, which iteratively deform the object's features to fit to an example of that object in a new image. The technique outlined by Cootes et al. [1] constrains the shapes by the Point Distribution Model (PDM) to vary only in ways seen in a training set of labelled examples. The PDM determines which shapes are plausible using the variance of the points in the training examples. This method can apply itself to many different problems, merely by presenting different training examples, while showing a compact representation of variance with few prior assumptions [1]. Our dataset contains both the original radiographs and mirrored versions. The mirrored radiographs were used to double the amount of training examples. The training set consist of the landmark of the first ten radiographs (and their mirrored counterpart), leaving the remaining four to be used for validation. Let's now first discuss how we pre-processed the provided landmarks.

### 2.1 Pre-processing landmarks

The incisors in the training set are represented by sets of 40 xy-coordinate landmark points. The points have to be aligned in the same co-ordinate frame for us to complete meaningful statistical analysis on their distribution. We can do this by applying a Procrustes analysis. The approach aligns each shape so that the sum of distances of each shape to the mean  $D = \sum |x_i - \bar{x}|^2$ . For the complete algorithm, we refer to Appendix A of the paper presented by Cootes et al. [1]. To implement the algorithm in Python, we make use of the procrustes function from the `scipy.spatial` library, which performs the similarity test for two points. We loop around this to perform procrustes on all of the landmarks for each tooth.

### 2.2 PCA

The landmark points of the incisors represent a high dimensional space. We can reduce the dimensionality of the data by applying a principal component analysis, projecting it into a lower dimensional space. PCA computes the main axes of variance in a dataset, arguing that all shapes can be represented by the following expression:  $\mathbf{z} = \bar{\mathbf{z}} + \mathbf{P}\mathbf{b}$  with Matrix  $P$  consisting of  $t$  principal components, and vector  $b$  the  $t$ -dimensional vector of parameters. The algorithm works by first computing the mean of the data. Consequently the covariance of the data is computed followed by the corresponding eigenvectors and eigenvalues. Every eigenvalue gives the variance of the data about the mean direction of the corresponding eigenvector [1]. We can then choose the  $k$  largest eigenvalues to keep in order to represent the data in a lower dimensional space, discarding part of the information in the original dataset.



**Figure 1:** A visualisation of the data post-procrustes analysis

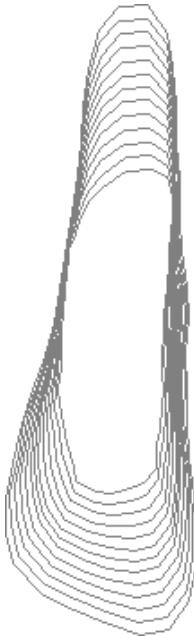
**Table 1:** Table showing the five largest eigenvalues of the PCA for each tooth

Tooth	$\lambda_1$	$\lambda_2$	$\lambda_3$	$\lambda_4$	$\lambda_5$
1	6.1725e+000	4.3783e-031	4.3790e-047	4.9372e-094	4.1791e-156
2	4.4580e+000	8.8818e-016	2.3484e-031	4.9372e-094	4.1791e-156
3	4.3760e+000	4.7395e-032	4.9372e-094	4.1791e-156	0
4	5.9104e+000	3.5639e-017	4.8071e-031	0	0
5	2.1963e+000	0	0	0	0
6	1.7047e+000	0	0	0	0
7	1.6979e+000	1.1390e-031	2.4686e-094	2.6612e-110	2.0896e-156
8	2.1927e+000	1.1093e-031	5.0069e-063	5.3976e-079	4.2381e-125

As we look at table 1, we can see that only one principal component has a large eigenvalue, letting us believe that most of the variance on a tooth resides in the first component. Therefore we only keep that component as a representation of the variance on the trained model. The effect of this first component can be seen in figure 2 for the upper left tooth.

### 3 Pre-processing of Dental Radiographs

The radiographs are inherently noisy data. The image contains much more than just teeth, including flesh, bones, braces and more. Before we fit our model on them, it would be wise to pre-process them to clearly define edges. The first filtering algorithm we decide to apply is the median filter as demonstrated in this paper by Barathi [2]. This kind of filter is very effective against a particular type of noise called salt and pepper noise. The particular algorithm makes use of a 5x5 kernel, setting every pixel to its median value in its neighbourhood. The result of this filter is shown in



**Figure 2:** A visualisation of the variance of the first component of the PCA

figure 4. The main benefit of this filter is its edge preservation properties. Applying it does not reduce our ability to detect edges later on. It is not very effective however in more spread out noise. For this reason we apply a bilateral filter with a 9x9 kernel in the next step. Bilateral filtering combines the idea of a weighted filter kernel with a better version of outlier rejection [3]. The weights are based on a Gaussian distribution and also preserve the edges in the original radiograph. The result of this transformation is shown in figure 5. For the final step in pre-processing, we take a look at three different filters: a (3x3) Scharr filter, a (3x3) Sobel filter and a (5x5) Sobel filter. The Sobel filter computes an approximation of the gradient of the image intensity function. An edge in an image occurs when the gradient is most extreme. The Sobel operator makes use of gradient direction in the x and y direction to detect these edges. The Sobel filter is not entirely symmetric. Scharr looked into optimizing this property by minimizing weighted mean angular error in a Fourier domain. This results in an approach that is consistent with symmetry constraints. The results from application of these filters are shown below in 8, 9 and 10.

## 4 Model Fitting

In this section we discuss how we fit the model generated in section 1 to the data we get from the processed image from section 2. This is done in 2 stages, first finding a good initial estimate for the model's position and



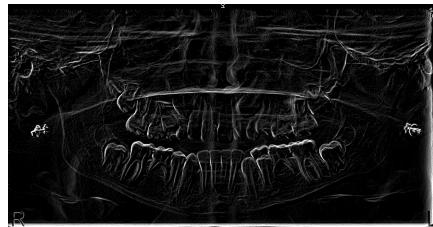
**Figure 3:** The original radiograph



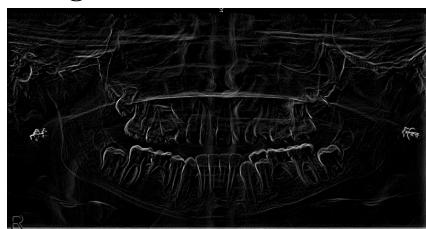
**Figure 4:** After median filter



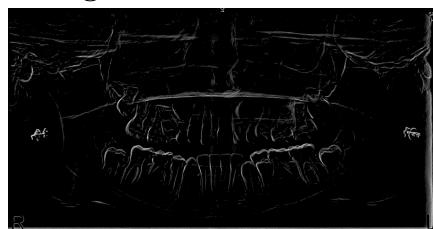
**Figure 5:** After bilateral filter



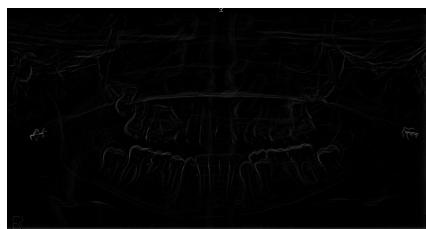
**Figure 6:** After scharr filter



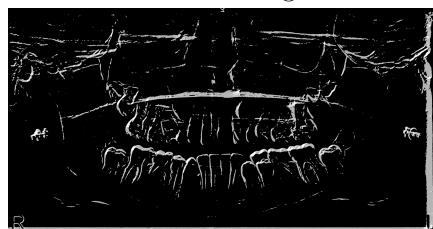
**Figure 7:** Scharr filter after normalization



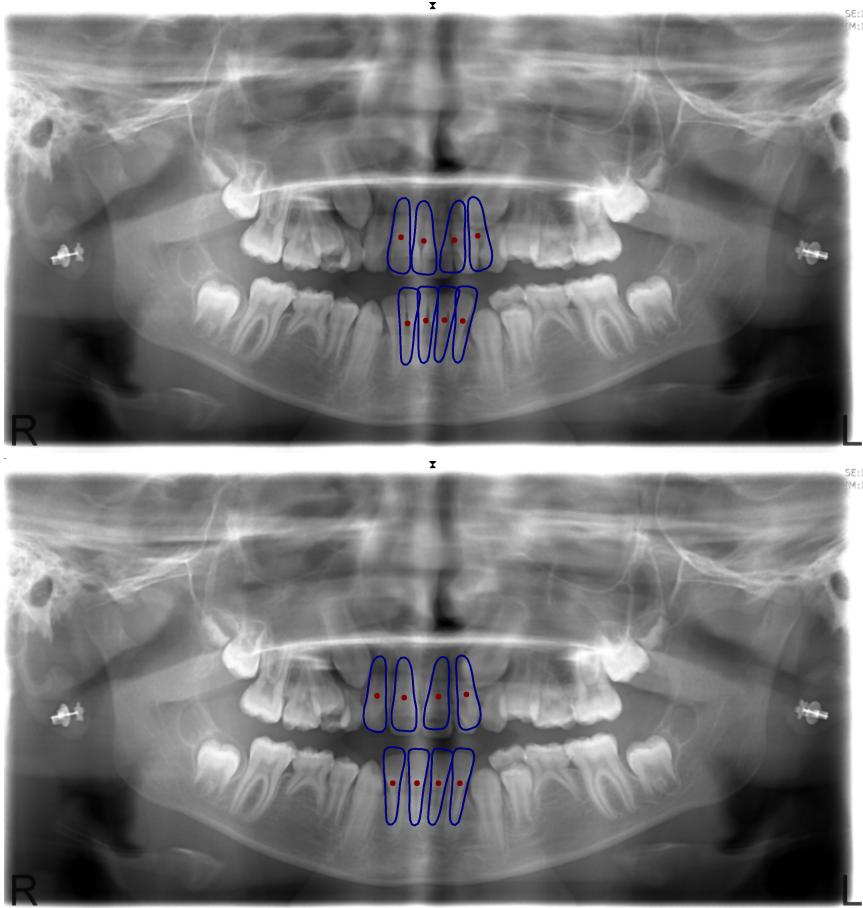
**Figure 8:** Scharr filter after thresholding



**Figure 9:** Sobel 3x3 filter (after thresholding)



**Figure 10:** Sobel 5x5 filter (after thresholding)



**Figure 11:** Initial estimate of the model’s position based on automated initialization (top) and manual initialisation (bottom)

second iteratively updating the model to better fit the data.

#### 4.1 Initial estimate

Finding a good initial estimate for the model’s position in the image has been proven to be crucial for finding a good border for the teeth. In the initial version of the model fitting code, this initial estimate is determined manually to ensure the estimate is relatively close to where the teeth is located. For each tooth, a reference point for the corresponding model is chosen so that the outline of the model is relatively close to the outline of the tooth. Later on an attempt was made to automate this process by approximating the initial position of the model for a tooth as the geometric average over all landmarks of that tooth. The result of both methods can be seen in figure 11.

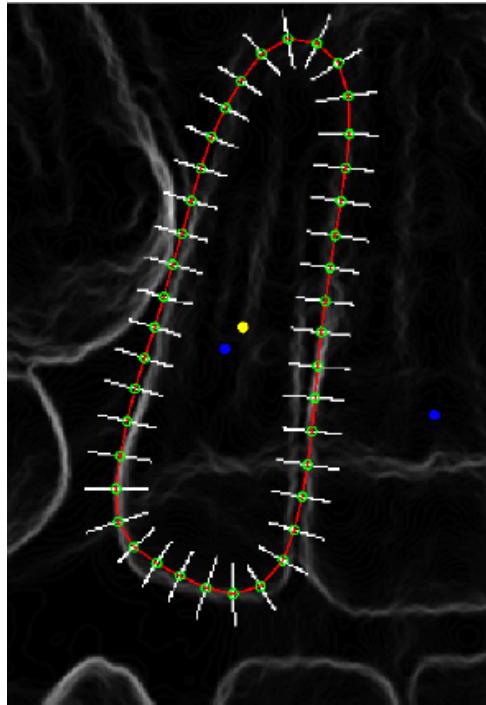
## 4.2 Iteratively updating the model

Finding the model that fits to the image the best is an iterative process that consists of three steps. The first step consists of finding a better position for the model by looking at the image. Each point on the outline of the model searches for a better location in the image by looking for a location in the image for which the edge detector yields a higher response. These new locations are a representation in which direction the model wants to change to better fit the image. To prevent the model from deforming too much at once, only locations perpendicular to the outline are considered. This is illustrated by figure 12.

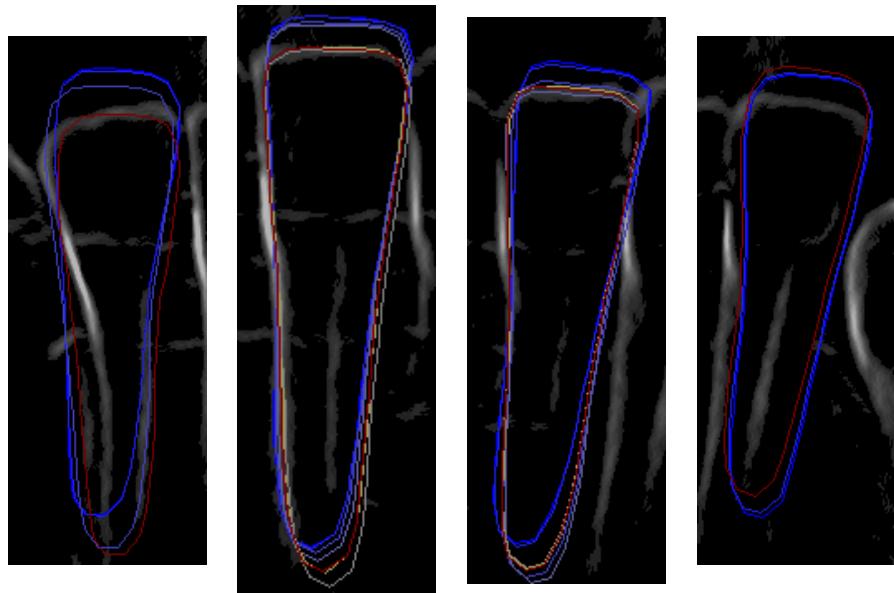
In a second step this desire to move to the newly found positions is met by aligning the model to these new positions by translating, rotating and scaling the current model. However this leaves discrepancies between the model and the new desired positions. To reduce these we allow the model to deform a bit in the directions in which the variances are the largest, as calculated by the principal component analysis on the training data. To prevent the model from deforming too much, we limit the maximum deviation of the model in regards to average model in each direction ( $b_i$ ) to lie between -1.25 and 1.25 times the standard deviation ( $\sim\sqrt{\lambda_i}$ ). For the details behind these computations we refer to the paper by Cootes et al. [4].

In a third step the parameters that control the outline of the model, such as the reference point of the model, scale and angle of the model and the amount that the model deviates from the mean model, are adjusted using the results from the previous step. The new outline of the updated model is calculated. Steps one till three are repeated using this new outline until the updated model exhibits a large deviation or until a number of iterations is reached. If the maximum number of iterations is reached the process is most likely to have converged although this is not guaranteed.

Another stopping criterion was considered, namely if a certain percent of points on the outline, probably 90% or higher, does not find a better position except its current position, then the process is stopped since it has most likely converged. We didn't opt for this criterion for the simple reason that it might preemptively stop due to a local optimum or not enough evidence for the points to move to a new position. We rather have it keep trying to improve the few points that want to improve to maybe find the global optimum or to move the model such that more points might reach the evidence of an edge. A visualization of this process can be seen in figure 13.



**Figure 12:** Visualization of the direction along better positions for the points outlining the model are sought



**Figure 13:** Visualization of the iterative process on the bottom row teeth. (blue = start, dark purple = 1 iter. , purple = 5 iter. , light purple = 10 iter. , gray = 25 iter. , yellow = 50 iter. ,red = end result)

**Table 2:** Average RMSE for radiographs 11 and 13

	Radiograph 11	Radiograph 13
Manual initialization	14.7885	19.4184
Automatic initialization	37.9285	72.8007

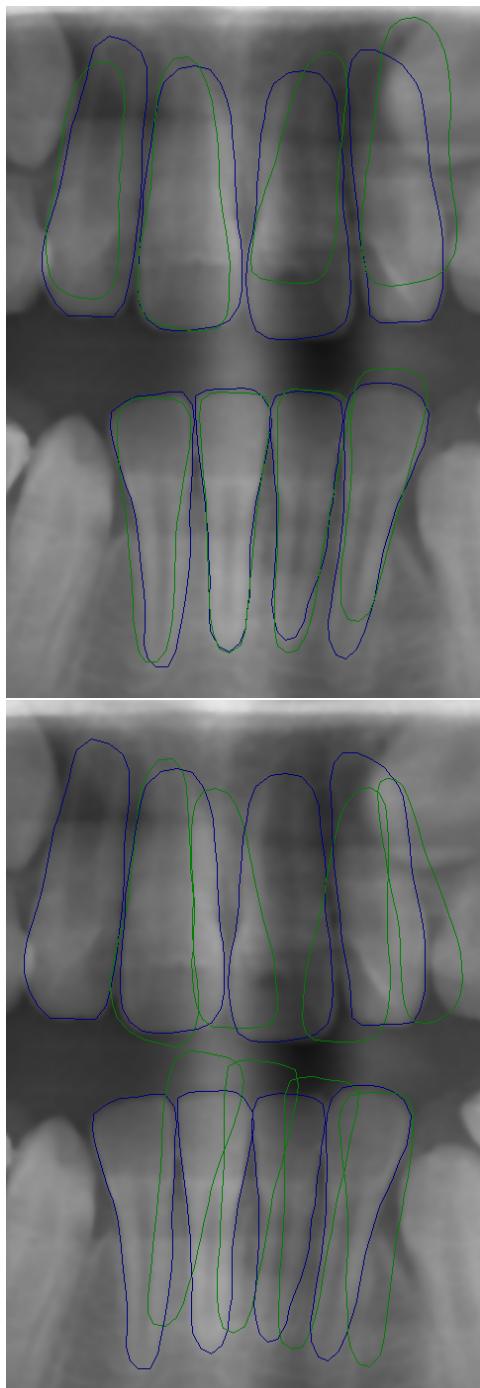
## 5 Evaluation of Results

From the results, illustrated by radiographs in figures 14 and 15, we can see that we get a decent approximation for the teeth's outline when using the manual initialisation. However when using the automated initialisation the results are worse due to the poor initialisation that the automatic initialisation does. This conclusion is confirmed when we look at the root mean squared error, shown by table 2. The RMSE is lower for the manual initialisation than for the automatic initialisation. This is most likely caused by high variations in the reference points for each tooth in the different radiographs.

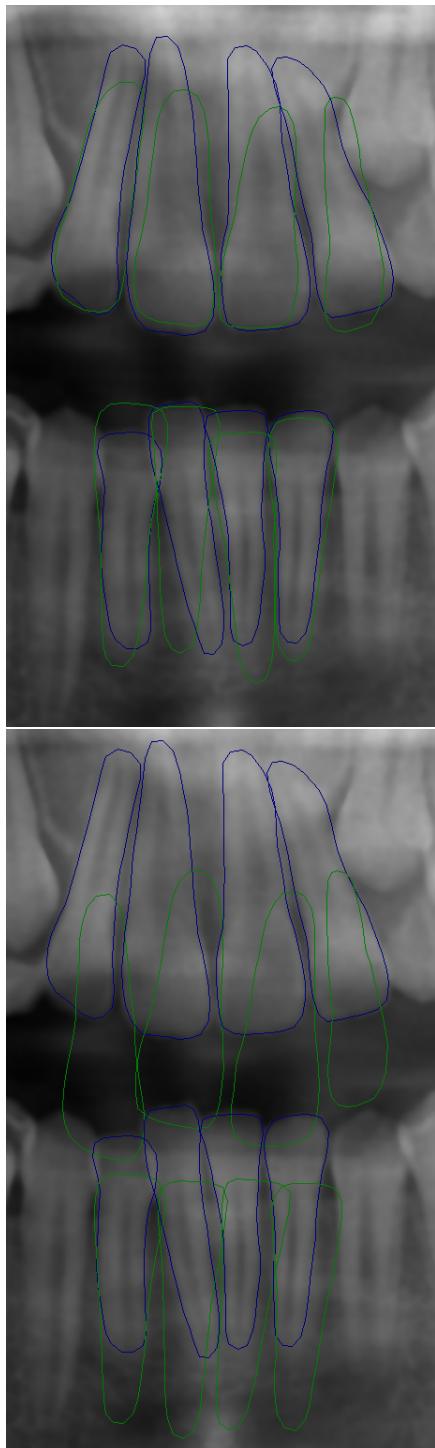
A better approach for automatic initialisation probably will be achieved by using a model that is based on the structures that arise on a higher level in the image to find an initial estimate for each tooth. Examples of these structure are the location of the jaw split to be used as a vertical reference on where the teeth are located and the vertical gaps between the teeth to help determining the horizontal position of the initial estimate for each tooth.

The quality of the image after preprocessing also determines the quality of our solution. Ideally the preprocessed image only contains the edges of the teeth we want to find and other edges that might arise from other structures in the image are filtered out. Another problem with this is the lack of edges for some parts of the teeth in the preprocessed image making it hard to find a good location for the points situated in that area of the tooth.

Another problem in our solution arises from the lack of multiple meaningful principal components. In our PCA only one component is responsible for most of the variation occurring in the shape of the teeth. Maybe by expanding our training set with more examples, we cover more variation on the shape of the teeth since our model can only adapt to variations it has encountered in the training set. Therefore making it possible to outline teeth whose shape varies in a different direction than our training set.



**Figure 14:** The result of the program on radiograph 11 using manual initialisation (top) and automatic initialisation (bottom)



**Figure 15:** The result of the program on radiograph 13 using manual initialisation (top) and automatic initialisation (bottom)

**Table 3:** RMSE for radiographs 11 and 13 using a manual initialization

Incisor	RMSE #11	RMSE #13
1	14.4442	16.2569
2	8.4498	21.4621
3	31.4428	27.6805
4	28.2760	28.9529
5	7.8246	13.4187
6	2.4289	13.7557
7	6.0858	23.6386
8	19.3552	10.1820

## 6 Conclusion

In this paper we attempted to create an incisor segmentation model to be used on dental radiographs. We started off by performing a procrustes analysis on the data to reshape it in order to define our active shape model based on the provided landmarks. Consequently, we performed PCA to reduce the size of the dataset and find the main axes in variance. The radiographs were pre-processed using different filters including median filters, bilateral filters, Scharr and Sobel filters. This was done to reduce the noise in images and ensure a smooth edge detection. Then we try to adapt our model to find the outline of the corresponding tooth in the image by looking at the pre-processed image and update the parameters of our model such as the location, rotation and scaling and allowing it to deform within reasonable boundaries.

In general the manual initialisation results in a decent approximation for the outline of the incisors in contrast to the poor results we get from the automatic initialisation. The approach suggested in the evaluation section should be explored. Others things to keep in mind is to expand the training set so it becomes more robust against more variation and maybe try to find a pre-processing process that leaves us with more cues about the location of the teeth and less irrelevant information such as other structures in the image (braces, etc.). Maybe other approaches can deal with these problems better and therefore should be explored, e.g. deep learning.

## References

- [1] T. Cootes, *An introduction to active shape models*. Retrieved (2017, June 25) from [http://www2.compute.dtu.dk/courses/02511/docs/asm\\_overview.pdf](http://www2.compute.dtu.dk/courses/02511/docs/asm_overview.pdf)
- [2] B. Barathi *Effective Filtering Algorithms for Enhancing Mammogram Images*, 2014, Computer and Communication Engineering, Tamilnadu College of Engineering
- [3] D. Gries, F. Schneider *Computer Vision: Algorithms and Applications*. 2010
- [4] T. Cootes et al. , *Active shape models - their training and application*, Department of Medical Biophysics, University of Manchester, 1994.