Object Segmentation using ASM and Good Features

Isaak Lim

Contents

Abstract

In this report we examine Object Segmentation using Active Shape Models. For this we will present the original approach and further improvements. The advances presented are the introduction of the Gray-level Appearance Model and a method to handle the problem of converging to local minima when training and segmenting with Active Shape Models. We also present an application of Active Shape Models in the field of Medical Image Processing and discuss the merits of the different presented approaches.

1 Motivation

As computers become more powerful over time, the possible applications of software to enhance and support daily life also grows. This is also true for the fields of computer vision and machine learning. With the increasing amount of data generated by hardware or software it is becoming harder to extract useful information from it. Segmentation in the domain of computer vision strives to handle the extraction of useful information from images. Its aim is to identify and extract given objects and classes of objects from images.
While computer vision has some well known applications in the automotive industry, it can also be applied to solve problems in the medical field. A major point of interest is to support medical professionals in their evaluation of medical images and the subsequent decision making process.
Since the correct assessment of medical-related images (e.g. CT images) is often vital, physicians frequently ask for second opinions in cases where the analysis is not easy. Thus, it would greatly support their work if software were to identify interesting regions or objects in question. It would not only help physicians make informed decisions but also give warnings if possibly vital information was missed or neglected.
In recent years Active Shape Models have emerged as robust solution for supervised segmentation of images. Apart from being used to identify and segment objects in a production line setup, they can also be used in the context of medical image processing.
In this report we will first explain what Active Shape Models are and how they are constructed in Section 2. We start by giving a general overview in Section 2.1. Next, we will then examine some improvements and developments that were proposed in recent years in 2.2. We continue with an example of an application of Active Shape Models in medical image processing in Section 3. A discussion of the methods and results presented in Section 2.2 follows in Section 4. Finally we conclude this report in Section 5.

2 Active Shape Model Segmentation

As van Ginneken et al. [6] point out, segmentation can generally be approached from two sides. When taking the bottom-up approach the image structure is analyzed and described as a collection of low-level elements and their spatial relationship to each other. The top-down approaches try to build a high level model that describes the class of objects that should be identified in images and segmented correctly. As van Ginneken et al. note, the problem with bottom-up approaches is that image structure alone does not necessarily provide a meaningful segmentation.
Active Shape Models are a top-down approach as the name implies. The model interprets points on the contour of the corresponding object as a point distribution. The goal is to capture the variations of the shape of the object class during training. The model is then used during fitting by displacing the points on the boundary of the model until they match the shape of the object in a given image.

2.1 General Method

We will now give an overview of the general method used to generate and apply Active Shape Models as presented by Cootes et al. [2]. The general approach is to first train the model with a set of training images as explained in Section 2.1.1. The second step is to use this model during the fitting process for new images that we would like to segment, which will be presented in Section 2.1.2.

2.1.1 Training The Active Shape Model Segmentation with ASMs is supervised, which means that we first need to a set of training images to construct the models in question. This is done by manually placing points on boundaries or features of the objects in the set of training images that we would like to recognize.

Once the points have been placed, a number of representative points are chosen. Cootes et al. [2] name three types of representative points or landmark points that can be chosen. The first type are application dependent points, which means that they depend on the type of objects we would like to recognize. An example for this would be selecting the points that mark sharp corners. The second type are application independent points, which as the name suggest do not depend on the type of objects that have to be picked out, like curvature extrema along the boundaries of the shape defined by the placed points. The third and last type of landmark points are points that can be interpolated from the first two set of points. Taking the midway points on edges between the landmark points of the first two types, would be an example.
After acquiring the set of landmark points we have to align the shapes from all training images in order to build a meaningful model. Therefore, we have to find a transformation, namely a rotation, scaling and translation, that minimizes the weighted sum of squares of the distances between corresponding landmark points of the shapes. Once all shapes are aligned to a given shape we calculate the mean shape from the aligned shapes.
Next we normalize the mean shape, which means that we choose some default orientation, scale and origin. Finally we realign all shapes to the mean. We repeat this process of calculating the mean, normalization and realignment until convergence. The normalization is essential to ensure convergence, since otherwise we would not have any meaningful comparison measure of alignment. Van Ginneken et al. note that in some cases alignment is not desired and therefore omitted [6]. This can be if the shape of the objects only varies slightly within a given range.
Now that we have aligned the shapes we want to derive a model that describes these shapes. This is done by statistically analyzing the set of given shapes. We can describe each shape by the coordinates of its landmark points. Assuming that the shapes consist of n landmark points each, we can describe each shape as a vector of n 2-dim points.
Furthermore, we can specify each shape to be a point in $2n$-dim space. Since the landmark points are partially correlated the points representing the shapes form a cluster in $2n$-dim space. Cootes et al. make the assumption that the points roughly approximate an high-dimensional ellipsoidal object. Following the assumption it is now possible to describe this cluster in terms of a center and major axes. The center is simply the mean $\overline{x}$. In order to capture the possible variations of the shape we calculate the deviation from the mean

$$dx_i = x_i - \overline{x} \tag{5.1}$$

where x_i is the shape vector. The deviations allow us to build a covariance matrix

$$\Sigma = \frac{1}{k} \sum_{i=1}^{k} dx_i dx_i^T \tag{5.2}$$

where k is the number of shapes extracted from training images.
Now we can perform PCA on the covariance matrix Σ in order to calculate the major axes defining the ellipsoidal object. Naturally, the eigenvector corresponding to the largest eigenvalue gives us the longest axis, along which the greatest variation can be found as visible in Figure 5.1. The eigenvalues of Σ are the variance of the shapes or the squared standard deviation from the mean. We can approximate the ellipsoid by choosing the l eigenvectors that cover the most variation.
It is now possible to model possible variations in the shape via linear combination of the eigenvectors of Σ. In order to keep the variations reasonable, Cootes et al. propose to clamp the scaling b_i of each eigenvector e_i to 3σ in the linear combination. This can also be described in terms of the eigenvalues of Σ as follows:

$$-3\sqrt{\lambda_i} \leq b_i \leq 3\sqrt{\lambda_i} \tag{5.3}$$

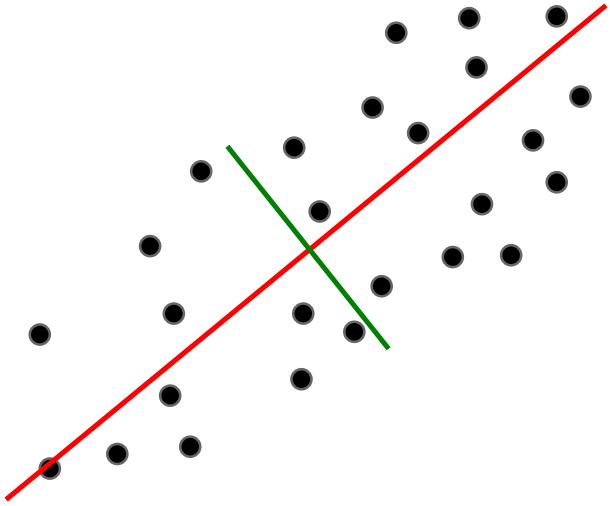


Figure 5.1: Illustration of the Principal Component Analysis in 2D. The red line represents the first Principal Component (eigenvector to the largest eigenvalue) and the green line represents the second Principal Component (eigenvector to the second largest eigenvalue)

2.1.2 Fitting The Active Shape Model Assuming the shape we are searching for is visible in the image, we now need to deform our ASM in such a way that it matches the shape. The boundary of the object we want to segment is given by the displaced landmark points of the ASM.
While there are several methods and approaches to training and fitting the ASM we need an initial estimate of the position of the landmark points in the image. Once the points are placed, we then iteratively transform the landmark points. This transformation is repeated until the position of the landmark points does not change significantly anymore. By examining the image structure around the landmark points in question we can determine if we should continue displacing the points.
Ideally this would result in landmark points that are placed on the boundary of the object in the image we are segmenting. However, this hugely depends on the quality of the model we have trained. Possible approaches will be discussed in more detail in Section 2.2.

2.2 Advanced Approaches

In this section we will mostly focus on the works of van Ginneken et al. [6] and Brunet et al. [1], which expand on the general method.
A first idea for fitting a model, might be to simply move each landmark point along one of the major axes computed with the PCA for the ASM or even along the normals of the boundary. However, the problem here is that the previous model is based on the assumption that there is a linear relationship between landmark points. This is why van Ginneken et al. propose an approach, where a gray-level appearance model is constructed in addition to the ASM and non-linear kNN-classifiers are used for training ASMs and then segmenting new images [6].

2.2.1 The Gray-level Appearance model In order to illustrate the proposed concept we will give an overview of what a gray-level appearance model is. In essence, a gray-level appearance model describes the image structure around a given landmark point. By calculating this model from the training images, we can then displace the landmark points of the mean model during the fitting process for a new image in such a way that the image structure around the transformed points is most similar to those of the model.
If we imagine the landmark points to be connected by edges as shown in Figure 5.2, the gray-level appearance model describes the slice through the landmark point along the normal defined by the edges to it's neighboring points in this case.

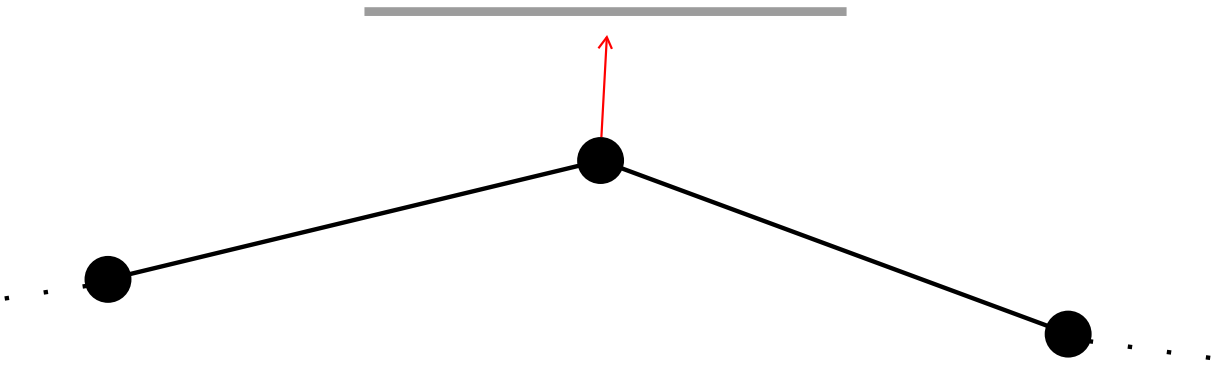


Figure 5.2: Illustration of landmark points (black dots) connected by edges. The red arrow represents the normal at one landmark point.

Along the normal we can sample an image k times in each direction, resulting in $2k+1$ samples overall for each landmark point. Since we want to describe the structure around landmark points in terms of the possible boundaries of the shape for our model, we need gradient information. From now on we will refer to the captured image structure around each landmark point as the profile.
Cootes and Taylor proposed to use normalized first derivates of the structure or the profile around the landmark point in question [3]. The mean $\overline{g}$ of these normalized first derivatives of profiles and the covariance matrix Σ_g describing the relationship between the normalized first derivatives profiles, composes the gray-level appearance model. g_i denotes an i-th profile for the i-th landmark out of the set of all landmarks. This allows us to compute the Mahalanobis distance of a profile of a displaced landmark g_i to the model during the fitting process as follows:

$$(g_i - \overline{g})^T \Sigma_g^{-1} (g_i - \overline{g}) \tag{5.4}$$

We can now compute the shape by minimizing the Mahalanobis distance.
However, this approach by Cootes and Taylor assumes that the landmark points and indeed their profiles are distributed normally. This is where van Ginneken et al. propose a different procedure to build the gray-level appearance model, since they argue that non-normal distributions of profiles occur frequently in practice [6].
Van Ginneken et al. point out that the best possible location to move the landmark point in question to during fitting, is the point where the profile of that location is “inside” the object we want to find on one side and “outside” on the other side. Since we would like to segment images of varying resolutions, van Ginneken et al. propose to use multi-scale Gaussian derivatives when constructing the gray-level appearance model. Therefore, we do not sample the original image we want to perform segmentation on, but instead first filter the image by convolution with derivates of bivariate Gaussians. In practice the first and second derivates are used.
It is now possible to describe the image structure or profile around each landmark points with a histogram of the intensity values of the filtered image. The first moment of the histogram gives us the mean value, while the second moment specifies the variance from the mean. It has to be noted that finding good parameters for the standard deviation σ of the Gaussians and the size of the window that the histogram covers for every conceivable image is not easy.

2.2.2 Training We will now present the proposed approach to training by van Ginneken et al. [6]. Since multi-resolution segmentation has to be supported, the task is to find the best set of features for each profile for all given resolutions as mentioned above. This means the set of features that enable the correct identification of the corresponding landmark in a given image. The features found in a profile for each landmark point can be described as a feature vector. We want to train a classifier such that on examining such a feature vector describing a sample of the profile around a landmark point we can classify it as “inside” or “outside” an object. Van Ginneken et al. arbitrarily define the landmark points themselves to be ”inside” the object.
In order to train the classifier, we have to validate its predictions. Van Ginneken et al. propose to use half of the image training set for training and the other half for validation. As a classifier 5-Nearest-Neighbor classifier is used. The idea is that when a new sample has to be labeled, its 5 nearest neighbors are

examined, which are already classified samples. A majority vote is then used to determine the class of the new label. So if 3 out of 5 nearest neighbors belong to class 'A', the new sample will also be assigned to class 'A'. An example of this is shown in Figure 5.3.

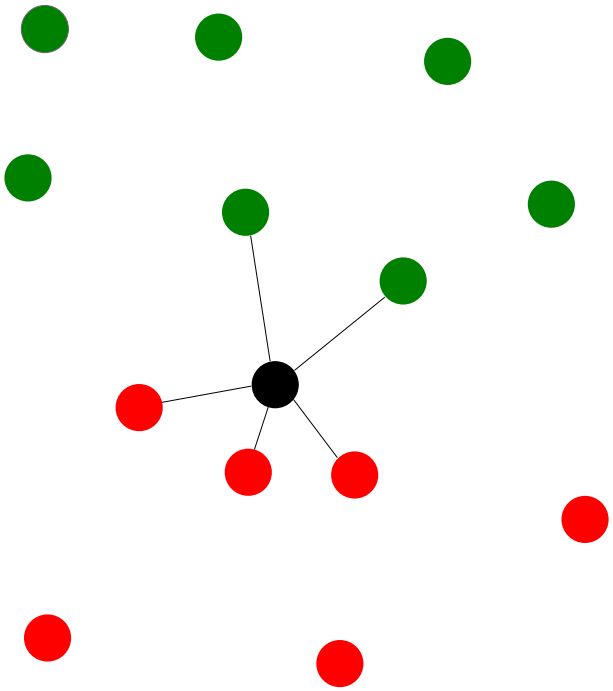


Figure 5.3: Illustration of the 5-NN classifier. The red and green dots represent two classes. The new sample is represented by a black dot. Its five nearest neighbors are connected to it by black lines. Since the majority of the 5 nearest neighbors belong to the red class, the new sample would also be labeled as red.

Van Ginneken et al. use feature vectors of size 60 and therefore introduce weighting to voting. The weight for each vote is $exp(-d^2)$, where d is the euclidean distance. We will further discuss the merits and disadvantages of this approach in Section 4.
In order to reduce the amount of features used to describe the model for performance reasons, while minimizing the loss of information van Ginneken et al. use sequential feature forward selection to select a given number of features out of the complete feature set for each landmark. Sequential feature forward selection is an algorithm that sequentially adds features that maximize some criterion function. While van Ginneken et al. do not explicitly state what criterion function they use, an example could be the Bhattacharyya distance function.
This function can be used to measure the similarity between two distributions. In the case of feature selection we select features that are most dissimilar to the other features in order to get a good coverage of the complete set of features for each grid of samples around each landmark point. Afterwards, sequential feature backward selection is applied. This means that features are removed from our subset if they improve the prediction accuracy. Somol et al. further expand on this method [4].

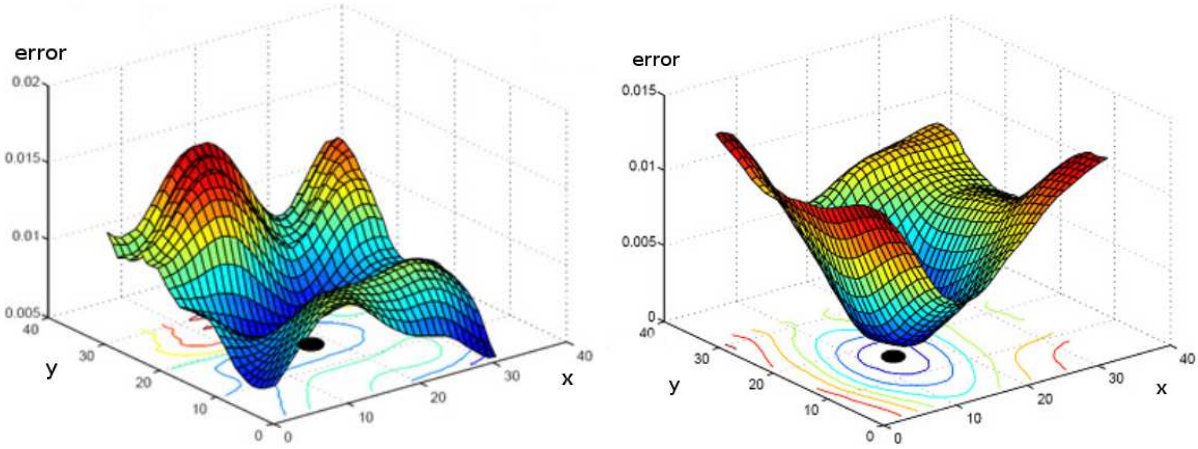
2.2.3 Fitting Now that we have trained the classifier, we can segment new images containing the object we have trained our models for. Van Ginneken et al. propose to initialize with the mean shape and start at the lowest level of resolution. The points on the profiles are used as input for the kNN-classifier. Unlike the approach presented in Section 2.2.1, we do not try to minimize the Mahalanobis distance in Equation 5.4, but the objective function

$$\sum_{i=-k}^{-1} p_i + \sum_{i=0}^{+k} (1 - p_i) \tag{5.5}$$

for all $2k+1$ points on each profile g. p_i stands for the probability of the i-th point being inside or outside the object we want to segment computed by the classifier. We then move the landmark points to the

points that minimizes this function. The index i denotes the index of each point on the profile running from $-k$ to k. The landmark point is positioned at the index $i = 0$. The probability 0 means that the point is outside the object and the probability 1 means that the point is inside the object. Therefore, minimizing the objective function for all profiles means displacing the landmarks in such a way that one side of the profiles with the indices of the points running from $-k$ to -1 are "outside" and the other side with indices of the points running from 0 to $+k$ are inside the object.
The two steps of evaluation and the movement of landmark points is repeated a set amount of times. Afterwards, these to steps are repeated for all higher resolutions. This results in the segmentation of a given image, where the shape of the object we are looking for is defined as the edges between the moved landmark points.

2.2.4 Handling Local Minima In this Section we will present the problem of converging to local minima in the context of ASMs and the proposed solution by Brunet et al. [1]. In order to train the model the image structure around the landmark point has to be considered as explained in Section 2.2.1. However, instead of using a slice throught the landmark here we use a 2D region or patch around the landmark as the profile for the gray-level appearance model.



(a) Error surface of a bad profile [1]

(b) Error surface of a good profile [1]

Figure 5.4: Examples of the error surface of good and bad regions around landmarks [1]

If we use PCA during training and pick l eigenvectors as the basis to describe a shape, we essentially reduce the dimensionality of the point cloud distribution. In the same way we can use a PCA of the feature vectors of the profiles in order to make an efficient evaluation and comparison of the image structure in the image we want to segment and our trained model possible. For every profile of each landmark point we can then calculate the reconstruction error for each point in the region as a measure of quality. The reconstruction error is the distance between the original value at that point and the linear combination of the l eigenvectors or any subspace that approximates the value at that point.
As visible in Figure 5.4(b), good features are regions around landmark points that have one local minimum instead of several as shown in Figure 5.4(a). If we have several local minima in the region we might move our landmark point to a minima which is not optimal during the fitting process. Therefore, a subspace needs to be found that minimizes the amount of local minima of the error surface around the landmark. We also want to dispose of the landmarks whose regions are not well suited for training our model.
Brunet et al. propose an approach to solve this problem by defining a criterion which makes it possible to select good regions around landmarks and introducing a supervised PCA method, which allows learning a good error surface. Let us first discuss the criterion by which landmarks and their regions are evaluated. Brunet et al. define the optimal placement of the landmark during fitting as minimizing:

$$\min_{a_1,a_2} \frac{||d_i^{x+a_1,y+a_2} - \mu - Bc_i||_2^2}{||d_i^{x+a_1,y+a_2}||_2^2} \tag{5.6}$$

a_1 and a_2 are translation parameters with regard to the positions x and y. $d^{x+a_1,y+a_2}$ are the feature values of the profile around the position $(x + a_1, y + a_2)$. B is the basis for an appearance model such as

the gray-level-appearance model given by PCA for example, while c contains the coefficients of the linear combination of the basis vectors. Finally, μ is the mean of the appearance model. The minimization is done by searching for the best parameters for a_1, a_2 and c. The results of this search result in the error surface as seen above.
In order to describe the properties of the error surface we count the number of local minima, consider the distance between the global minimum of the error surface and the expected position of the landmark and finally the distance between the global minimum of the error surface and the closest minimum. These properties are influenced by varying the size of the profile and the amount of basis vectors we use for our appearance model.
Brunet et al. compute local PCA with varying size of the profile and amount of basis vectors for each landmark independently. They found that the more basis vectors are taken into account the more local minima exist. Furthermore, increasing the size of the profile also increases the number of local minima. However, as expected the density of local minima decreases with increasing size of the profile. After evaluating these factors for each landmark Brunet et al. found a good trade-off to be profile sizes of 30 x 30 pixels and a basis, which has 80% of the original dimensionality of the appearance model.
Using these parameters on all the landmarks and their profiles gives us a criterion, which allows us to compare profiles and rank them. Brunet et al. do this by describing each profile as follows:

$$\mu_{profile} = \mu_{1_{profile}} + \mu_{2_{profile}} \tag{5.7}$$

$$\sigma_{profile} = \frac{\mu_{1_{profile}}\sigma_{1_{profile}} + \mu_{2_{profile}}\sigma_{2_{profile}}}{\mu_{1_{profile}} + \mu_{2_{profile}}} \tag{5.8}$$

Here $\mu_{1_{profile}}$ and $\sigma_{1_{profile}}$ are the mean and variance of the distance between the global minimum of the profiles to the expected position of the landmark. $\mu_{2_{profile}}$ and $\sigma_{2_{profile}}$ are the mean and variance of the number of local minima of the profiles. The profiles are then ranked according to how well they minimize $\mu_{profile}$. This allows us to select a given number of the best landmarks and their profiles for training our model.
As mentioned above we would also like to find a subspace of features for our ASM that minimizes the amount of local minima. The best case would be if the error surface were to resemble an inverted Gaussian as visible in Figure 5.5.

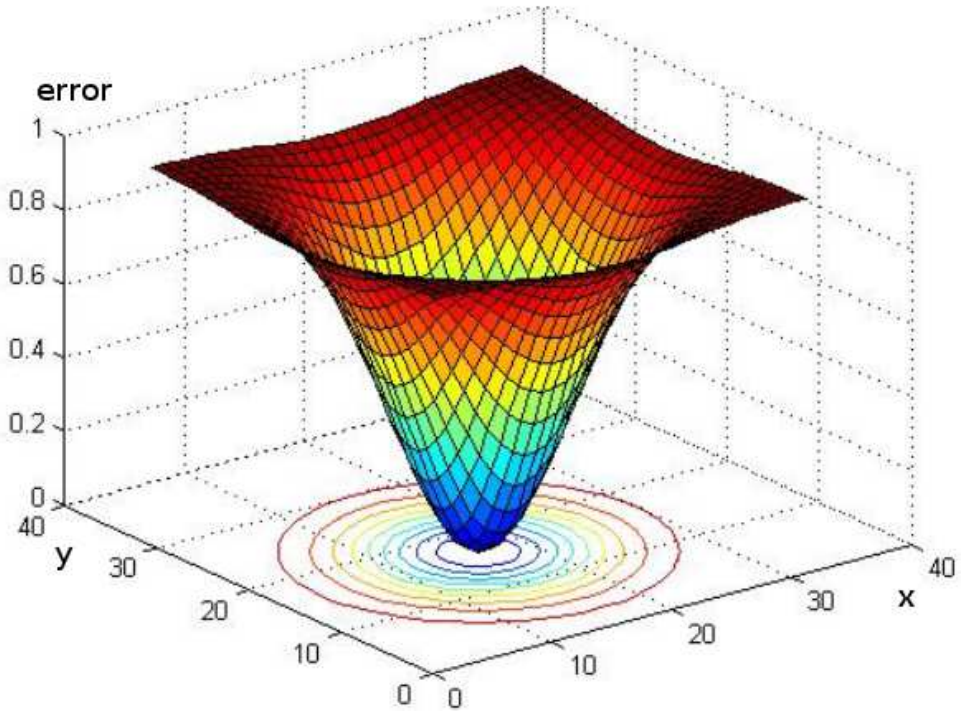


Figure 5.5: Inverted Gaussian response function [1]

Therefore, Brunet et al. evaluate the error surface of each profile by comparing it to an inverted Gaussian

$$E = \sum_{i=1}^{n} \sum_{(x,y)} (\alpha_i r_{x,y} + o_i - \frac{||d_i^{(x,y)} - \mu - Bc_i||_2^2}{||d_i^{(x,y)}||_2^2}) \tag{5.9}$$

Here $r_{x,y}$ is the response of the inverted Gaussian at the position (x, y). α_i is the scaling factor of the inverted Gaussian and o_i is the translation factor. Minimizing this equation means finding a subspace for the appearance model where the minimum is at the position of the landmark and the response in the profile around it is similar to that of an inverted Gaussian. However, minimizing the equation by finding a suitable subspace B is not easy. Brunet et al. state that there is no closed form solution for the minimization. Therefore, they use the gradient descent method in order to minimize to above equation. The idea behind gradient descent is that a multivariate function $F(x)$ decreases fastest if we assume that we start at $F(x_0)$ and move x_0 towards the negative gradient of $F(x_0)$ as shown in Figure 5.6.

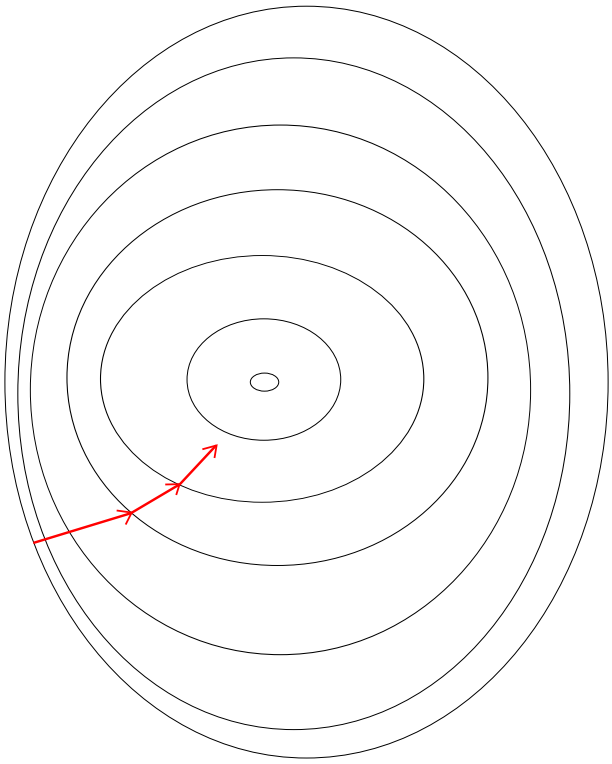


Figure 5.6: Illustration of gradient descent on contour lines of a multivariate function

Obviously $F(x)$ has to be defined and differentiable around x_0. In terms of the Equation 5.9 we can specify the gradient as follows:

$$G = \frac{\delta E}{\delta B} \tag{5.10}$$

The $n + 1$ iteration of the gradient descend method for the subspace B would thus be:

$$B^{(n+1)} = B^{(n)} - \eta G \tag{5.11}$$

For the translation and scaling factors o_i and α_i a closed form solution exists. The method proposed by Brunet et al. alternates between apply gradient descent to B and solving for o_i and α_i. After each iteration the new subspace basis B is orthonormalized, which means that each basis vector is normalized and made to be orthogonal to all other basis vectors.
Finding an optimal η in Equation 5.10 as stepsize can be done by apply line search, which minimizes an objective function. In this case

$$E(B - \eta G) \tag{5.12}$$

is used. Line search applies a similar idea to gradient descend, where the objective function is minimized by searching for a minimum along the descend direction.

Brunet et al. note that the minimization of Equation 5.9 is prone to converging to local minima. For this reason they start gradient descend from several initial points and then pick the solution with the smallest error. Furthermore, B is initialized with a PCA. By selecting landmarks with good profiles surrounding them and finding a subspace with good features for training the ASM, Brunet et al. show that they are able to achieve better results during fitting than the standard PCA approach. We will further discuss the results in Section 4.

3 Applications In Medical Image Processing

After having introduced ASMs above, we now want to present an application in the field of medical image processing. As an example we have picked *Automated 3-D Segmentation Of Lungs With Lung Cancer In CT Data* by Sun et al [5]. Since our main focus in this report is on ASMs we will not go into great detail concerning the method presented in this paper.

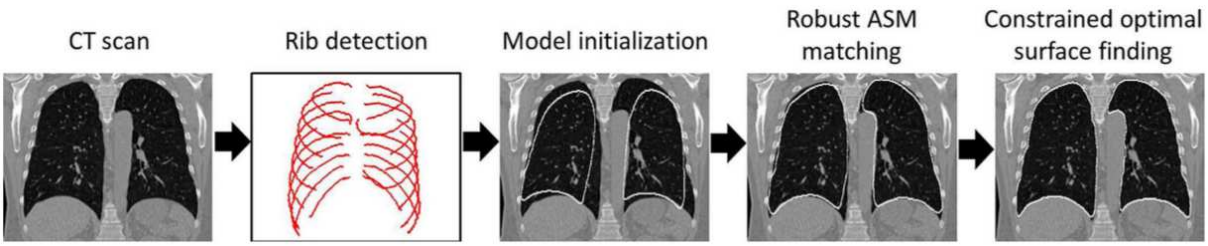


Figure 5.7: Pipeline of the approach [5]

The goal is to have a robust segmentation lungs with diseased tissue. The segmentation is achieved by using a ASM approach visualized in Figure 5.7. As a training set healthy lung CT scans are used. Since we have volumetric data we can compute 3D segmentation instead of a 2D one. The landmark points of each slice through the volume make up the points on the surface of each lung in the training set. It is then possible to extract a triangle mesh from the surface by using marching cubes. Next a point distribution model is build from the mesh points by using a similar method to the one described in Section 2.1.
In order to have a good initial position of the ASM during fitting, Sun et al. propose to use the rib cage as a point of reference. The rib cage itself is found by examining the density values of the volumetric data. The standard ASM fitting approach would be to search along the surface normal at the landmark positions and evaluate some cost function which considers the density values along the normal. This process would be iterated until convergence as explained in Section 2.1.
However, Sun et al. point out that the density of diseased lung tissue is much higher than that of normal lung tissue. This causes the fitting to converge to a solution where to updated landmark points do not lie on the surface but at the transition of the diseased and the normal lung tissue. For this reason they propose to train a model which recognizes normal lung shape patterns. The model can then be used during fitting to identify outliers and reject solutions via a voting mechanism.
Finally, a smooth surface is found by using a maximum flow algorithm where the graph edges reflect the properties of the volumetric data.

3.1 Results

Examples of the results achieved with this method are shown in Figure 5.8.

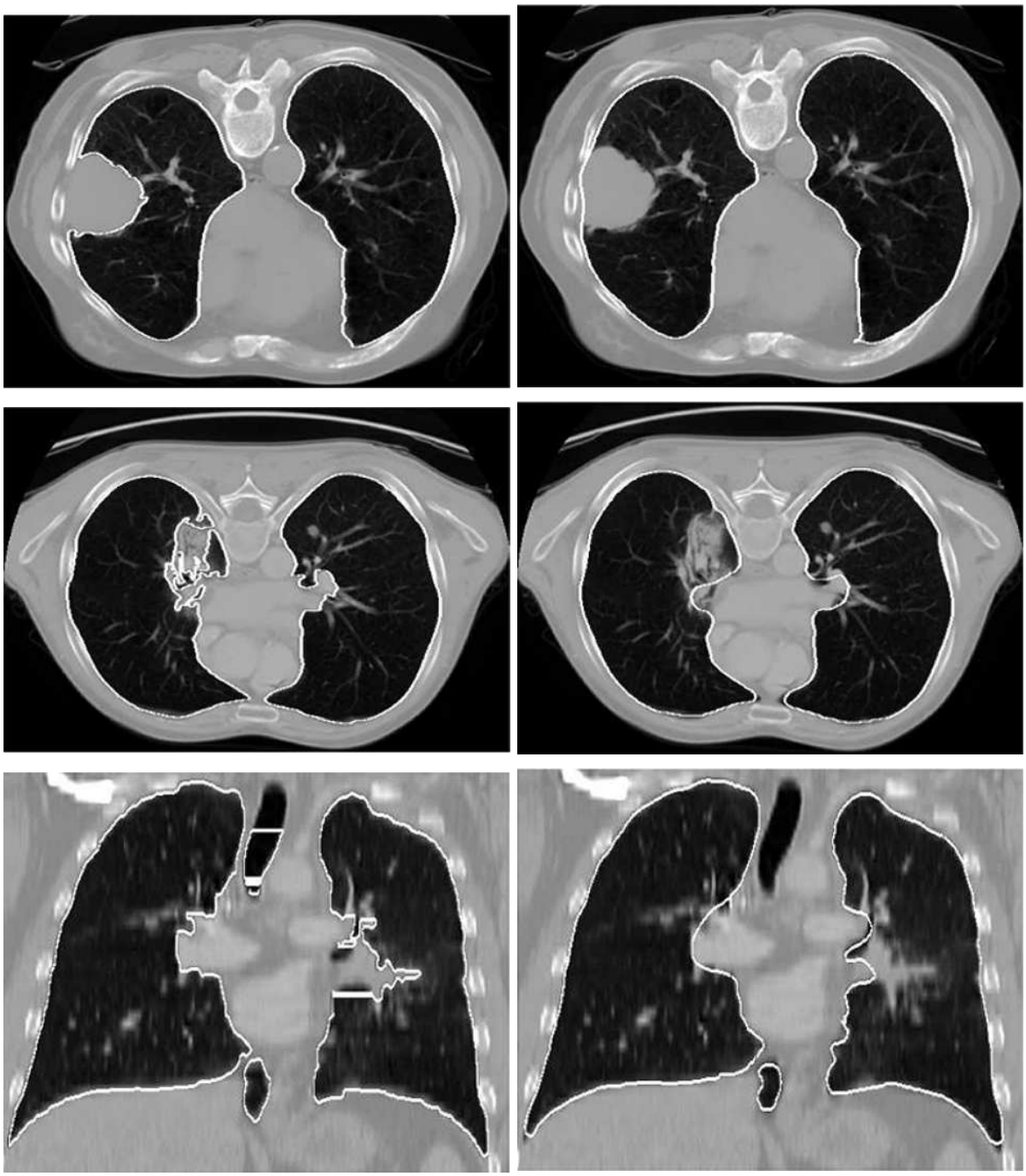


Figure 5.8: The images show different slices through CT data. The segmentation in the left column shows the manually done defined reference, while the right column show examples of the segmentation results for the same set of images [5]

Sun et al. note that they managed to produce robust segmentation of both normal and diseased lungs, while the standard ASM method, which produces a least squares optimization during fitting has problems with non-normal lung structures, since it is sensitive to outliers.

(a) Standard method [5] (b) Proposed method [5]

Figure 5.9: Differences in the results show that in this case the standard method causes the segmentation to leak into the gas filled colon, while the proposed robust ASM method still performs well [5]

4 Discussion

Since we mostly focused on the proposed approaches to ASMs by van Ginneken et al. [6] and Brunet et al. [1] in Section 2.2, we will now discuss possible shortcomings and results in this Section.

4.1 Active Shape Model Segmentation With Optimal Features [6]

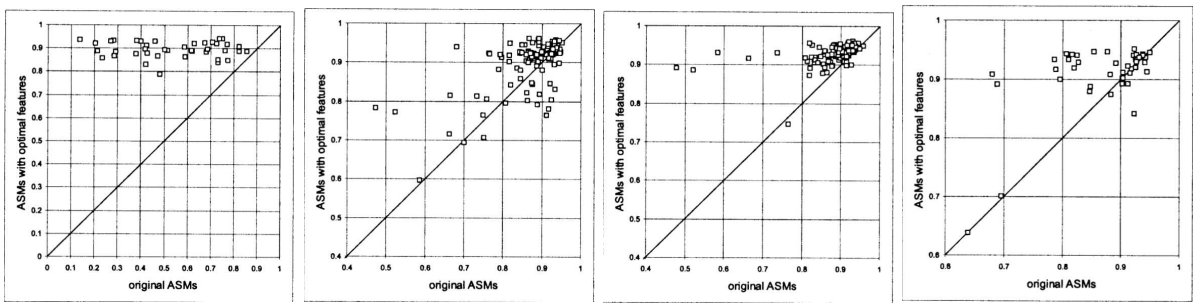
While van Ginneken et al. state that it is possible to use any other classifier for their method, the reasons for the choice of the kNN-classifier are not made clear. Furthermore, selecting an optimal k is not easy. Van Ginneken et al. do not specify why they chose $k = 5$ in their paper. It would have been interesting to see the results of their method with different k and how they compare to the presented results.
We assumed that a weighting with $exp(-d^2)$ was chosen for the votes in the kNN-classifier in order to handle the "*curse of dimensionality*" when comparing 60-dimensional feature vectors. As explained in Section 2.1.1, d is the Euclidean distance between two feature vectors. Van Ginneken et al. do not address why their weighting is constructed in the way it is. A high number of dimensions makes it harder to define neighborhood in the traditional manner since Euclidean distances grow more and more alike. This is due to the fact that the differences in each dimension sum up over the total number of considered dimensions. Using a different distance function or an approach that explicitly take the high number of dimensions into account for the nearest neighbor search might improve the quality of the results.

4.1.1 Results We will now present some of the results by van Ginneken et al. In order to evaluate their method for segmentation they measured the overlap:

$$\Omega = \frac{TP}{TP + FP + FN} \tag{5.13}$$

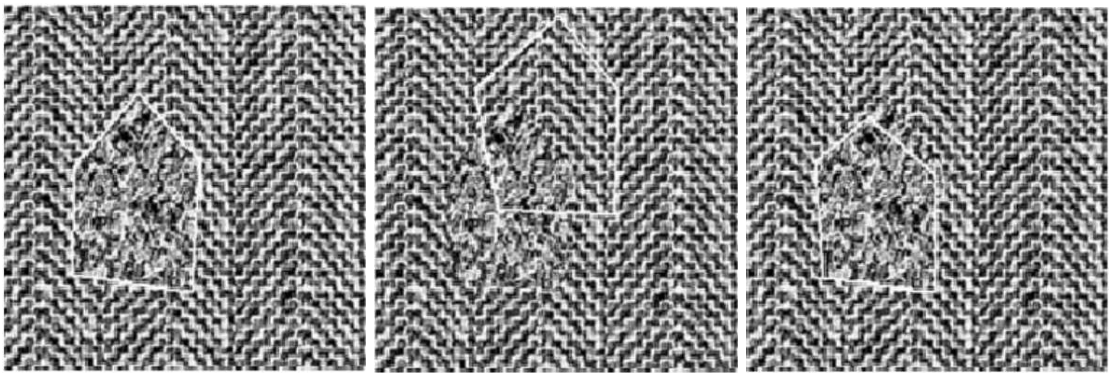
TP stands for *true positives*, FP for *false positives* and FN for *false negatives*. A perfect result would thus be $\Omega = 1$. The experiments where done by selecting half of the respective image set randomly for training and the other half for validation.
As visible in Figure 5.10, van Ginneken et al. mainly compared the original ASM method to their method with "optimal" features. Figures 5.11 and 5.12 show examples of segmentation results on different images. Figure 5.11(a) and 5.12(a) are manually defined ground truths.

We can see that their method generally outperforms the standard ASM method in almost all cases. By choosing "optimal" features the approach proposed by van Ginneken et al. handle different types of texture better.



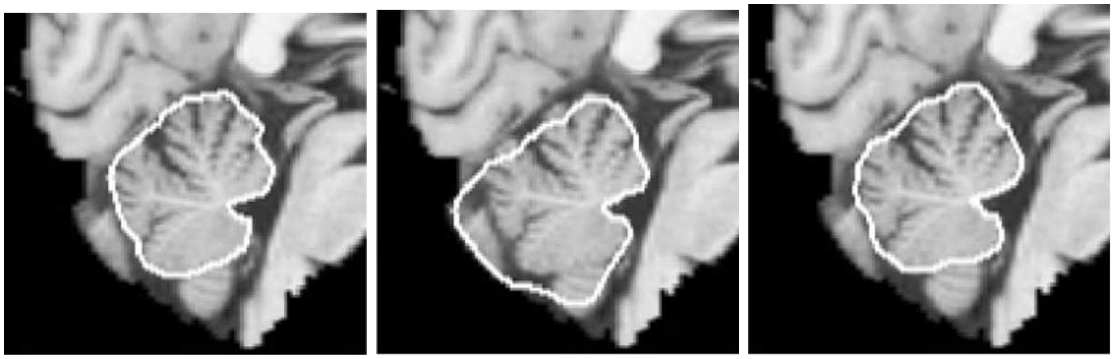
(a) House (b) Left Lung Field (c) Right Lung Field (d) Cerebellum

Figure 5.10: Scatter plots for segmentation experiments. The overlap measure Ω for the original ASM method (x-axis) is plotted against Ω for the ASM method with optimal features (y-axis) [6]



(a) True Shape (b) ASM (c) Optimal ASM

Figure 5.11: Example results for the segmentation of generated textured images with a house shape [6]



(a) True Shape (b) ASM (c) Optimal ASM

Figure 5.12: Example results for the segmentation of MRI images of a cerebellum [6]

4.2 Learning Good Features for Active Shape Models [1]

Brunet et al. clearly state the problem of local minima within the context of good features for ASMs. Their approach to find a good subspace with gradient descent and line search that minimizes the Equation 5.9 however is also prone to local minima as Burnet et al. note. They offer a solution to this problem

by initializing the subspace B with a PCA solution and starting gradient descend from different random points. However this method is not further evaluated. How does the number of different starting points correlate with the quality of the subspace? It would also be interesting to see how other methods like the Newton method for optimization, which converges faster than gradient descend compares in terms of quality and performance. It has to be noted however that the problem of global optimization is very difficult to solve.

4.2.1 Results Brunet et al. did a set of experiments in detecting facial features in order to evaluate their method of selecting good profiles. They compared the mean squared error of an ASM built with 68 landmarks to one built with 34 landmarks. Both ASMs were constructed using their method of selecting the best regions as seen in Section 2.2.4. The ASM with 34 landmarks was built by choosing the best profiles for each part of the face as shown in Figure 4.2.1.
An example of the fitting results with the two ASMs can be seen in Figure 5.14. As visible in Figure 5.15, Brunet et al. managed to half the landmarks and still achieve similar results, which allows an boost in performance without loss of quality.

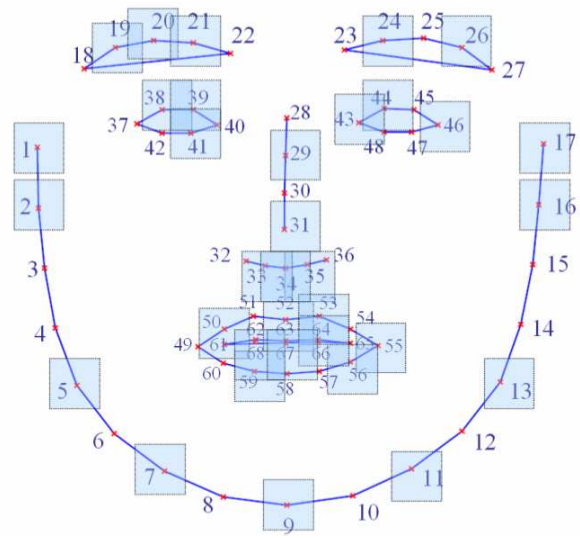
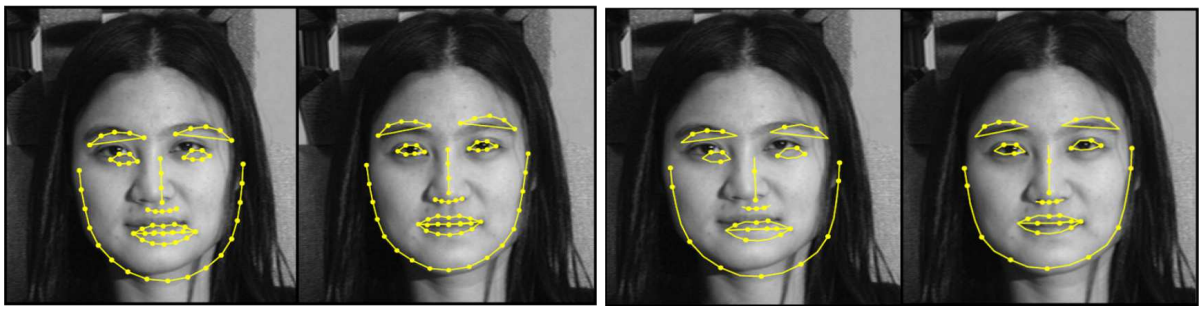


Figure 5.13: Best profiles for each part of the face [1]



(a) 68 landmarks (b) 34 landmarks

Figure 5.14: Example of facial feature detection. On the left in 5.14(b) and 5.14(a) is the initial ASM configuration and on the right the final result is shown [1]

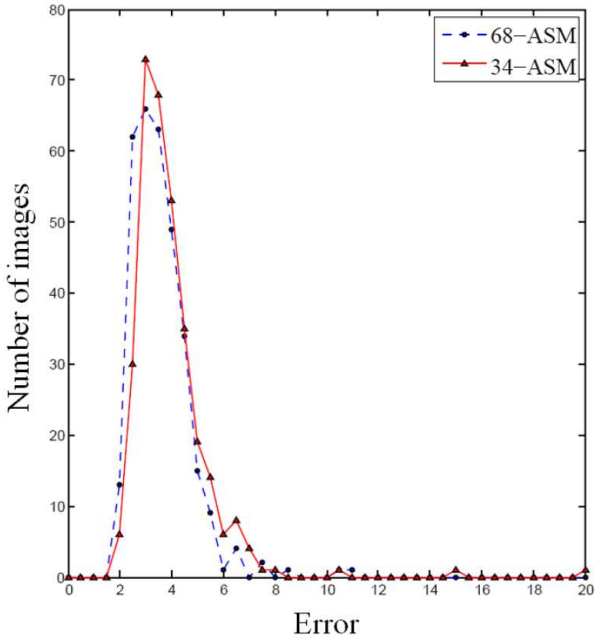


Figure 5.15: Error histogram of the ASM with 68 landmarks and the ASM with 34 landmarks [1]

An illustration of the improvement of learning good subspaces for profiles is shown in Figure 4.2.1. Here the error surface of two profiles from Figure 4.2.1 are shown. On the left the error surface resulting from standard PCA is visible, while on the right the error surface corresponding to the learned subspace optimizing Equation 5.9 is shown.
Brunet et al. also compared the quality of their method in comparison to using a PCA model. 200 images were used to train ASMs with 68 landmarks. 100 image were then used to test the quality of detection. The results are shown in Figure 4.2.1. As can be seen, the method by Brunet et al. outperforms the PCA model in terms of quality.

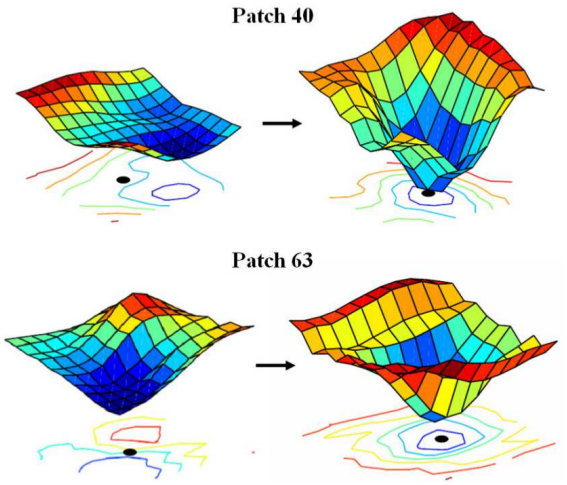


Figure 5.16: Error surface before and after learning good features [1]

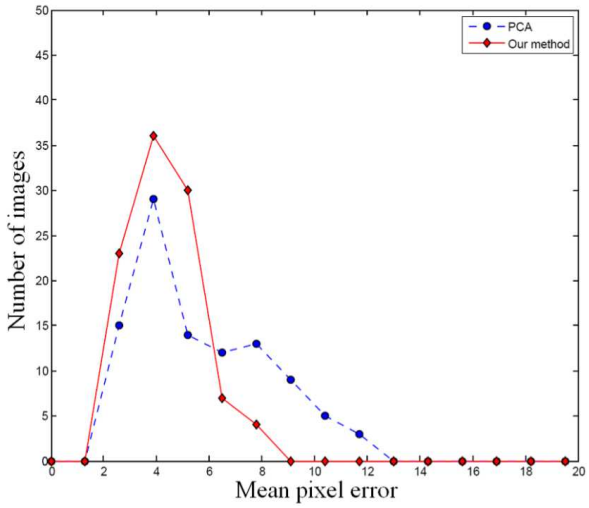


Figure 5.17: Error distribution comparing PCA and the method proposed by Brunet et al. [1]

5 Conclusion

In this report we introduced Active Shape Models and their extensions. We also discussed their application in the context of lung segmentation. While the standard method does not perform well in all cases, the general idea is robust. This is due to the fact, that it relatively easy to work with the existing standard model and extend it as needed.
As shown in Section 3, ASM-based techniques are a valid solution to provide segmentation in the medical field. Furthermore, the current results with regards to learning which features should be tracked presented in Section 4.2 are very promising. With additional advancements in computer vision and machine learning applied to the medical field, patients and physicians are bound to benefit even further from software-aided image analysis in the near future.

References

[1] N. Brunet, F. Perez, and F. De la Torre. Learning good features for active shape models. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 206–211. IEEE, 2009.

[2] T.F. Cootes, C.J. Taylor, D.H. Cooper, J. Graham, et al. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995.

[3] T.F. Cootes, C.J. Taylor, et al. Statistical models of appearance for computer vision. *World Wide Web Publication, February*, 2001.

[4] P. Somol, P. Pudil, J. Novovičová, and P. Paclık. Adaptive floating search methods in feature selection. *Pattern recognition letters*, 20(11):1157–1163, 1999.

[5] S. Sun, C. Bauer, and R. Beichel. Automated 3-d segmentation of lungs with lung cancer in ct data using a novel robust active shape model approach. *Medical Imaging, IEEE Transactions on*, 31(2):449–460, 2012.

[6] B. Van Ginneken, A.F. Frangi, J.J. Staal, B.M. ter Haar Romeny, and M.A. Viergever. Active shape model segmentation with optimal features. *Medical Imaging, IEEE Transactions on*, 21(8):924–933, 2002.