TECHNIQUES FOR STATISTICAL SHAPE MODEL BUILDING AND FUSION

Constantine Butakoff
(Kostantyn Butakov)

The cover image design by
Constantine Butakoff
Milton Hoz de Vila

ISBN 978-90-810592-0-6

Printed by Agpograf Impressors, Barcelona, Spain

PHD THESIS

TECHNIQUES FOR STATISTICAL SHAPE MODEL BUILDING AND FUSION

Constantine Butakoff
(Kostantyn Butakov)

UNIVERSIDAD DE ZARAGOZA
Instituto de Investigación en Ingeniería de Aragón

BIOMEDICAL ENGINEERING
Joint PhD Program from Universidad de Zaragoza and Universitat Politécnica de Catalunya

Thesis Director: **Dr. Alejandro F. Frangi**
Universitat Pompeu Fabra, Barcelona, Spain

The research described in this thesis was carried out at the center for Computational Imaging and Simulation Technologies in Biomedicine(CISTIB), UPF, Barcelona. This work was partially funded by the grant from Dirección General de Investigación, Innovación y Desarrollo de de Aragón (B177/2004); grants TIC2002-04495-C02, FIT-390000-2004-30 and TEC2006-03617/TCM from the Spanish MEC; FIT-360005-2007-9 from the Spanish Ministry of Industry; CDTI CENIT-CDTEAM; grants IM3/G03/185 and FIS/2004/040676 from ISCIII; BioSecure European Network of Excellence (IST-2002-507634) funded by the European Commission.

Abstract

In the present thesis we address the problems of building and combining active shape and appearance models. The models are one of the widespread tool for object modeling and segmentation with a shape and texture prior. When these models are employed several problems can arise:

1. expensive training process (in terms of time and memory requirements)
2. a training set of images with the delineated object (usually manually) is required
3. a high degree of uncertainty of the delineations (due to the presence of noise) including unfeasibility of manual delineations in 3D.

To overcome these problems we propose:

1. A framework for weighted fusion of multiple active shape or active appearance models based on eigenspace combination. Such combination strategy can be treated as a linear interpolation of the models. The benefit of the fusion is that the combined model can represent any object, which can be assumed to be a linear combination of the objects corresponding to the fused models. In other words, if an object has a number of typical appearances (different face expressions or different face poses, or different cardiac pathologies), it is possible to choose the most representative ones and assume any other to be a linear combination of the representative set. Then the combined model can be used to accurately segment the object in question and weights can be used for classification to determine, which representative appearance is closer. The possible applications of this framework are: batch model construction, object

classification based on combination weights, reduction of training sets to only representative appearances.

2. A view-independent face segmentation algorithm based on the fusion of active appearance models. This algorithm can be used to segment any facial pose and also determine the pose angle using the estimated combination weight. Only the views corresponding to the extreme head poses and the frontal one are taken for training, all the other poses are assumed to be a linear combination of these. Estimation of combination coefficients through reconstruction error minimization allows finding the optimal combined model, which is more specific to the pose under consideration than a single model constructed for all poses.

3. Combination of computed tomography (CT) and synthetic ultrasound (US) and single photon emission tomography (SPECT) images to automatically learn shape variation and voxel intensity variation, where it is demonstrated how intensity information can be learned for the two modalities where the resolution or quality is too low to manually annotate the images, especially in 3D. In this case generation of synthetic images through realistic simulation of the imaging process allows learning the appearance for a given set of shapes (obtained from high quality CT scans).

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

When casting pebbles into water,
look at the ripples,
otherwise this activity
will be an empty amusement

(Kozma Prutkov)

1.1 Overview

It all started by a 5×5cm image of a three-month-old son of Russell Kirsch back in 1957. The ghostlike black-and-white photo marked the birth of the computer imaging as we know it today...

As digital images were penetrating every aspect of our lives, growing both in quantity and quality, the more and more important was becoming the question of not only improving image quality but also the question of automated image analysis. Today the problems of computer aided image retrieval and analysis are of utmost interest to the world market, security and healthcare. They encompass such areas as image denoising, reconstruction, extraction of imaged objects, analysis of appearance and behaviour of the extracted object, classification, and many, many more. The algorithms are being developed to efficiently analyze all kinds of data, from web searches to security related and medical, in the latter case allowing more efficient and rapid handling of critical events.

From the day computers started to be used to handle images, the complexity

of the tasks was rapidly increasing, accompanied by the increasing availability of computational power. While the earliest problems were essentially related to image enhancement and recognition, nowadays the focus has extended to a high level analysis of the scenes in images or events in videos. The world today has been drowned in information that requires processing and humans are not fast enough to do it by themselves. For example, in the analysis of hours of surveillance videos there is much effort to reduce human work to a minimum. Motion detectors are used to reduce the quantity of data to be analyzed. Guards of big buildings have to constantly monitor the output from many video cameras spread throughout the place. This work is very tedious and still it is difficult to simultaneously pay attention to all the cameras. If this still could be handled by a motion detector issuing an alarm, in the frequented areas like parkings, jewelry stores, or airports what needs to be detected is not human presence but rather suspicious activity, and much effort is being carried out in this area.

The basic building block of almost any high level scene analysis is the separation of any particular object form the rest of an image, or segmentation (note that segmentation also refers to the methods of separating an image into areas with similar characteristics). Once the object of interest has been extracted, it can be efficiently analyzed for patterns in its appearance end behavior. Of course extracting only one object, also eliminates the context in which it has appeared, so perhaps all the objects in the scene must be identified and separated from each other. But that is even more complex task and remains generally unsolved.

In the following two sections some light is shed on why model-based segmentation is interesting and where it is applied in the fields of biometrics and biomedicine.

1.2 Segmentation in facial biometrics

By far the most popular area here is the security. As the world terrorism is on the rise, it stimulated a lot of investments into the development of systems for human authentication and identification as well as classification of human activity. The areas of interest here are the analysis of human activity, which requires separation of the human body from the background and analysis of its shape in time, and verification or identification of a person by his face. The latter is typically done using a database of faces or by comparing to a photo carried by the person. Biometric passports with a photo on a chip is an example of this. For not requiring almost any user collaboration, facial biometric is probably the most favored (although one of the least reliable) identification approach. Very few technologies compete with faces in this aspect (among which voice and gait are probably the most important) and the technology is already being used or tested in many airports worldwide. The technology is still far from perfect, as can be seen from the attempts to inte-

grate facial biometrics to recognize owners of laptops (typically followed by articles in press about researchers being able to hack the system by showing it a photo). The latter is essentially because it is hard to distinguish a live person from a photo using a webcam. In these contexts, the segmentation of the facial shape probably will not be able to help detecting the liveliness of the face, but, if accurate, it helps improving the robustness of the recognition by eliminating background and helping analyze the shape of the face separately from its texture (as for example is done in Active Appearance Models). In other words, it is possible to get rid of unnecessary (for face recognition) face variations such as facial expressions.

Another field of interest is the facial expression and/or pose analysis. In his works, Darwin claimed that all people, regardless of race or culture, possess the ability to express some emotions in exactly the same ways through their faces. Only much later this was confirmed by Paul Ekman in a series of experiments [1–3], which revealed agreement both within and across cultures for six emotional expressions - anger, disgust, fear, happiness, sadness, and surprise. These data were the first systematic evidence for the universality of emotions and their expressions. This stimulated research in analysis of human emotions for both medical, marketing and simply better human-computer interaction. So, for example, in 2006 there was quite some hype about an "emotionally aware" computer, that uses a camera to capture images of the user's face, then determines facial expressions, and infers the user's mood. Since then it is being developed in University of Cambridge by Peter Robinson. Since human emotions usually are portrayed on the face even when we are alone (of course it does not work for everybody, like, for example, everybody knows that Chuck Norris has only two facial expressions: with and without the beard), it seems very attractive to be able to determine them with just a simple video camera, especially considering that today webcams are easily affordable and most notebooks come with a built-in one. So the typical approaches to facial expression recognition are based either on direct classification of images with the cropped face or classification of features extracted in a more sophisticated way. These features can be for instance parameters of a 2D or 3D face model adjusted to match the face in the image. These shapes can subsequently be analyzed for the presence of certain expressions. Additionally such model could provide information about the pose or gaze. The latter is becoming quite popular to help disabled people interact with computer using essentially their eyes (for instance, gaze tracking is used to move the mouse pointer and blinking - for clicks).

1.3 Segmentation in biomedicine, in cardiac imaging

Diseases of the heart and circulatory system (cardiovascular decease or CVD) are the main cause of death in Europe, accounting for over 4.3 million (2.0 million in the EU) deaths each year (48% of deaths) [4]. Many imaging techniques have

been developed to perform cardiovascular examinations. Ultrasound (US), single-photon emission computed tomography (SPECT), computed tomography (CT) and magnetic resonance imaging (MRI) are by far the most well-known and established techniques. Among them, cardiac ultrasound still remains the most ubiquitous cardiac imaging modality, with applications at the bedside and during interventions. At the same time, it is the best modality in terms of temporal resolution and the time spent on acquisition. In many cases this is the first modality used to analyze the patient's condition.

Cardiac multidetector computed tomography (MDCT) has established itself as the modality for assessing the structure of the coronary tree in vivo with simultaneous acquisition of the dynamic anatomy of the whole heart and great vessels with great spatial detail (0.5 mm isotropic voxels). Unfortunately, this modality still involves substantial radiation, which makes it less suitable when follow-up scans need to be performed.

Cardiac MRI provides an abundant source of detailed, quantitative data on heart structure and function. It is non-invasive and safe. It is able to provide high-quality functional information in any plane and any direction, meaning that it is possible to get views of the entire heart, irrespective of its orientation. Cardiac MRI has provided detailed information on 3-D ventricular shape and geometry, regional systolic and diastolic strain, material microstructure, blood flow, perfusion and viability [5].

As one can see there is an abundance of information obtained from the different modalities and many of them are complementary. So there is much interest in trying to combine those data in a single frame of reference. It can afterwards be used to build cardiac atlases or bio-mechanical and physiological models, personalizing them to every specific patient. All of these is being fostered by the rapid growth of accessible computational power. Model-based analysis tools allow calculation of the global function indices like left ventricular mass and volume. On the other hand they allow quantitative parametrization of regional heart wall motion which gives more insight into the wall dynamics and presence of local abnormalities. This also could provide means for statistical comparison of hearts drawn from different patient populations.

1.4 Contributions

This thesis is all about one of the most widespread landmark-based statistical shape representation methods coined *Point Distribution Model*s (PDMs) [6]. There are two popular modeling and segmentation methods connected to this representation: *Active Shape Models* (ASM) [6, 7] and *Active Appearance Models* (AAM) [8, 9]. Shortly after their introduction, these methods became very popular in many applications and several interesting contributions have been proposed in the literature to extend

or improve the original formulation [10–19]. These methods are particularly attractive for their simplicity, robustness and speed, and gained good acceptance in facial analysis and biometrics as well as in several medical image analysis applications.

In contrast to their execution efficiency, their training could require large representative datasets, which implies:

- The contours of the object of interest in the training set have to be delineated (manually or by some other automatic or semiautomatic means).
- The training can take a considerable amount of time. This is particularly the case of the AAM [8, 17], due to the costly estimation of large prediction matrices for parameter updates. Moreover the complexity grows with the size of the average face (because all the pixels have to be modeled). Another problem is construction of the model from huge training sets, due to a necessity to compute the covariance matrix and its eigendecomposition, the memory requirements can grow very fast.

This brings us to the contributions presented in this thesis:

1. A framework for combining multiple active shape or active appearance models, Chapter 2. This framework treats the models as a set of eigenspaces (defined by eigenvectors and eigenvalues) and proposes a framework for their combination with a possibility of assigning weights. Such combination strategy can be treated as a linear interpolation of the models. The benefit of doing the aforementioned combination is that the combined model can represent objects modeled by the two models and any intermediate object. In other words, if an object has a number of typical appearances (different face expressions or different face poses, or different cardiac pathologies), it is possible to choose the most representative ones and assume any other to be a linear combination of the representative set. Then the combined model can be used to accurately segment the object in question and weights can be used for classification to determine which representative appearance is closer. The possible applications of this framework are: batch model construction, object classification based on combination weights, reduction of training sets to only representative appearances.
2. A view-independent face segmentation algorithm based on the fusion of active appearance models, Chapter 3. With this approach we demonstrate how the combination of models can be used to segment any facial pose and also determine the pose angle using the estimated combination weight. Only the views corresponding to extreme head poses and the frontal one are taken for training, all the other poses are assumed to be a linear combination of these. Estimation of combination coefficients through segmentation error minimization allows finding the optimal combined model which is more specific to the pose under consideration than a single model constructed for all poses.

3. Combination of CT and synthetic US/SPECT images to learn shape variation and voxel intensity variation, Chapter 4 and 5. Finally, these chapters demonstrate how different cardiac imaging modalities can be combined to train a segmentation algorithm and adapt it to a specific imaging modality, which alone has insufficient quality for that task. In this case generation of synthetic images through realistic simulation of the imaging process allows learning the appearance for a given set of shapes (obtained from high quality CT scans).

CHAPTER 2

A Framework for Weighted Fusion of Multiple Statistical Models of Shape and Appearance

Abstract - *This paper presents a framework for weighted fusion of several Active Shape and Active Appearance Models. The approach is based on the eigenspace fusion method proposed by Hall, Marshal & Martin [20], which has been extended to fuse more than two weighted eigenspaces using unbiased mean and covariance matrix estimates. To evaluate the performance of fusion, a comparative assessment on segmentation precision as well as facial verification tests are performed using the AR, EQUINOX and XM2VTS databases. Based on the results it is concluded that the fusion is useful when the model needs to be updated online or when the original observations are absent.*

Adapted from C. Butakoff, A.F. Frangi. A Framework for Weighted Fusion of Multiple Statistical Models of Shape and Appearance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1847–1857, 2006.

2.1 Introduction

THIS article focuses on one of the most widespread landmark-based statistical shape representation methods coined *Point Distribution Model*s (PDMs) [6]. There are two popular modeling and segmentation methods connected to this representation: *Active Shape Models* (ASM) [6,7] and *Active Appearance Models* (AAM) [8,9]. Shortly after their introduction, these methods became very popular in many applications and several interesting contributions have been proposed in the literature to extend or improve the original formulation (*e.g.*, [10–12,14–16,21]). These methods are particularly attractive for their simplicity, robustness and segmentation speed, and therefore gained good acceptance in facial analysis and biometrics as well as in several medical image analysis applications. However, in contrast to their execution efficiency, their training from large datasets can take a considerable amount of time, particularly for AAM, due to the costly estimation of large Jacobian matrices. Whenever the model needs to be updated with new data, it has to be retrained using past and new observations. However, past observations may be no longer available, or the need to store all past observations may simply be impractical for on line model updating. Moreover the larger is the database of observations the more costly will be the process of recomputing the model and, consequently, an incremental model updating strategy could constitute a valuable alternative. In dynamic model learning it is useful to introduce mechanisms for "forgetting" past data [22,23]. This can be accomplished during model retraining by assigning lower weight to past observations than to new observations. To summarize, in this context it is important to be able to update the model based only on the past model parameters with a possibility to assign different weights to past and new data.

Taking a look at classical ASM and AAM, one may notice that these models are essentially eigenspaces. As defined in [20], an eigenspace of a set of observations is a quadruple consisting of mean, eigenvectors, eigenvalues and the number of observations. In particular an ASM has one eigenspace and a set of covariance matrices, and an AAM has three eigenspaces and two Jacobians. This leads to the hypothesis that, essentially, AAM and ASM fusion could be reduced to eigenspace fusion.

The solution that we propose is based on the eigenspace fusion framework introduced in [20]. Nowadays there are many applications where eigenspace construction and analysis are involved. These applications include classification, motion sequence analysis, temporal tracking, segmentation with statistical models, and many others, and there are already many works on eigenspace updating strategies in the literature. Reasons for eigenspace fusion are essentially the same as those mentioned above; the most important being, perhaps, the need to quickly and constantly update the eigenspace to keep it up-to-date with new available data. There are many methods for updating eigenspaces by one observation at a time [24–27] and for the fusion of already computed eigenspaces [20,28]. We use the method

proposed in [20] because it computes matrices of the smallest possible size thus minimizing memory requirements and computation speed. Among the newest publications it is worth to mention a work by Zhou et al. [29], where an eigenspace fusion framework is employed to efficiently track a shape, represented by a set of control points in a vector form [7], placed along object's contour. There, the authors derive a formulation of a Kalman Filter to track the shape, based on the fusion of eigenspaces described in [20].

The main contributions of this paper are a generalization of the eigenspace fusion algorithm introduced in [20] and an AAM/ASM fusion framework. Originally, its authors did not take into account weighting of fused eigenspaces, considered only fusion of two eigenspaces, and all the covariance matrix estimates were biased while it is common practice to use unbiased estimates. Therefore we generalized the algorithm to be able to perform weighted fusion of any number of eigenspaces, and modified it to use and compute unbiased estimates of covariance matrices. Then, taking this modified eigenspace fusion as a starting point, we thoroughly develop the framework for ASM and then AAM fusion. As a consequence, this paper is primarily concerned with deriving a theoretical framework for fusion, rather than its particular application to any specific area, and only a set of generic experiments are performed to evaluate the algorithm and to compare the performance of the fused model to the model constructed from the original observations.

The paper is organized as follows. Section 2.2 covers the problem of weighted eigenspace fusion, which is a generalization of the algorithm proposed in [20]. Sections 2.3, 2.4 and 2.5 introduce the fusion of Active Shape and Active Appearance Models. Section 2.6 summarizes the steps for fusing AAMs. Then, Sections 2.7 and 2.8 demonstrate the results and draw conclusions. Finally Appendices at the end of the article provide some additional information and derivations for the reader's convenience.

2.2 Weighted eigenspace fusion

Before we begin, let us note that matrices will be written in bold uppercase, vectors will be column vectors and written in bold lowercase, and that letters with normal typeface will denote scalars.

Let us consider M eigenspaces. Each i-th eigenspace is computed by Principal Component Analysis (PCA), applied to the set of N_i observations $\mathbb{X}_i = \left\{\mathbf{x}_{ij} | j = 1, \ldots, N_i\right\}$, each being an n-dimensional column vector, and is defined as a quadruple [20]:

$$\Omega_i = \left(\bar{\mathbf{x}}_i, \mathbf{\Phi}_i, \mathbf{\Lambda}_i, N_i\right), \quad i = 1, ..., M \tag{2.1}$$

where the n-vector $\bar{\mathbf{x}}_i$ is the mean of the observations, $\mathbf{\Phi}_i$ is a $n \times m_i$ matrix of eigenvectors, and $\mathbf{\Lambda}_i$ is a $m_i \times m_i$ matrix of eigenvalues (n and m_i have been determined

during construction of eigenspaces to be fused). Note that the number of rows of all $\mathbf{\Phi}_i$ is the same (*i.e.,* all the eigenvectors must have the same number of components). Each eigenspace is assigned a weight w_i such that $\sum_{i=1}^{M} w_i = 1$. These weights are used to change the influence of each eigenspace on the fused one. Without loss of generality we shall assume that all the weights are positive. Let

$$p_i = w_i \cdot \left(\sum_{j=1}^{M} w_j N_j \right)^{-1} \tag{2.2}$$

Introducing p_i we transfer the model weights w_i to the observation level, so each observation of i-th model has a weight p_i. Let us define the full observation set as $\mathbb{X} = \bigcup_{i=1}^{M} \mathbb{X}_i$, consisting of $N = \sum_{i=1}^{M} N_i$ observations, with its elements denoted by $\mathbf{z}_k \in \mathbb{X}$ in order to simplify the notation in several formulas (each $\mathbf{z}_k$ is equal to $\mathbf{x}_{ij}$ for some i and j).

The goal of fusion is to compute such an eigenspace $\Omega = (\bar{\mathbf{z}}, \mathbf{\Phi}, \mathbf{\Lambda}, N)$, using the information from Ω_i only, that it is equivalent to the eigenspace computed from the full set of observations $\mathbb{X}$. Here, $\bar{\mathbf{z}}$ is again a n-vector, $\mathbf{\Phi}$ is a $n \times m$ matrix, and $\mathbf{\Lambda}$ is a $m \times m$ matrix, where m is determined during the fusion.

Let us define the function $P\left(\mathbf{x}_{ij}\right) = p_i$, which can be thought of as a probability of observing $\mathbf{x}_{ij}$. Then the fused mean is

$$\bar{\mathbf{z}} = \sum_{k=1}^{N} P\left(\mathbf{z}_k\right) \cdot \mathbf{z}_k = \sum_{i=1}^{M} \sum_{j=1}^{N_i} p_i \mathbf{x}_{ij} = \sum_{i=1}^{M} N_i p_i \bar{\mathbf{x}}_i \tag{2.3}$$

Now let us denote the covariance matrices by $\mathbf{D}_i = \mathbf{\Phi}_i \mathbf{\Lambda}_i \mathbf{\Phi}_i^T$ and the fused covariance matrix by $\mathbf{D}$. The unbiased estimate of the fused covariance matrix (see Appendix I) is then :

$$\begin{aligned}
\mathbf{D} &= \frac{1}{1 - \sum_{k=1}^{N} P\left(\mathbf{z}_k\right)^2} \cdot \widetilde{\mathbf{D}} = \frac{1}{1 - \sum_{i=1}^{M} N_i p_i^2} \cdot \widetilde{\mathbf{D}} \\
\widetilde{\mathbf{D}} &= \sum_{k=1}^{N} \left(\mathbf{z}_k - \bar{\mathbf{z}}\right)\left(\mathbf{z}_k - \bar{\mathbf{z}}\right)^T P\left(\mathbf{z}_k\right) = \\
&= \sum_{k=1}^{N} P\left(\mathbf{z}_k\right) \mathbf{z}_k \mathbf{z}_k^T - \bar{\mathbf{z}}\bar{\mathbf{z}}^T = \sum_{i=1}^{M} p_i \left(\sum_{j=1}^{N_i} \mathbf{x}_{ij} \mathbf{x}_{ij}^T \right) - \bar{\mathbf{z}}\bar{\mathbf{z}}^T
\end{aligned} \tag{2.4}$$

Rewriting the expression for $\mathbf{D}_i$ according to the definition

$$\mathbf{D}_i = \frac{1}{N_i - 1} \sum_{j=1}^{N_i} \left(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i\right) \left(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i\right)^T = \\ = \frac{1}{N_i - 1} \left[\sum_{j=1}^{N_i} \mathbf{x}_{ij}\mathbf{x}_{ij}^T - N_i \bar{\mathbf{x}}_i \bar{\mathbf{x}}_i^T \right]$$

or

$$\sum_{j=1}^{N_i} \mathbf{x}_{ij}\mathbf{x}_{ij}^T = (N_i - 1)\,\mathbf{D}_i + N_i \bar{\mathbf{x}}_i \bar{\mathbf{x}}_i^T \tag{2.5}$$

Substituting (2.5) and (2.3) into (2.4) we obtain

$$\tilde{\mathbf{D}} = \sum_{i=1}^{M} p_i \left(\sum_{j=1}^{N_i} \mathbf{x}_{ij}\mathbf{x}_{ij}^T \right) - \bar{\mathbf{z}}\bar{\mathbf{z}}^T = \\ = \sum_{i=1}^{M} (N_i - 1)\,\mathbf{D}_i p_i + \sum_{i=1}^{M} N_i p_i \bar{\mathbf{x}}_i \bar{\mathbf{x}}_i^T - \\ - \sum_{i=1}^{M} N_i p_i \bar{\mathbf{x}}_i \cdot \left(\sum_{i=1}^{M} N_i p_i \bar{\mathbf{x}}_i \right)^T = \sum_{i=1}^{M} (N_i - 1)\,\mathbf{D}_i p_i + \\ + \sum_{i=1}^{M-1} \sum_{j=i+1}^{M} N_i N_j p_i p_j \left(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\right) \left(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\right)^T$$

Finally

$$\mathbf{D} = \frac{1}{1 - \sum_{i=1}^{M} N_i p_i^2} \cdot \left[\sum_{i=1}^{M} (N_i - 1)\,\mathbf{D}_i p_i + \right. \\ \left. + \sum_{i=1}^{M-1} \sum_{j=i+1}^{M} N_i N_j p_i p_j \left(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\right) \left(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\right)^T \right] \tag{2.6}$$

We wish to compute the eigenvalues and eigenvectors that satisfy $\mathbf{D} = \mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Phi}^T$. The method of solution is as in [20]: construct an orthonormal basis set $\mathbf{Y}$ that spans all the eigenspaces; use $\mathbf{Y}$ to derive an intermediate eigenproblem, whose solution provides eigenvalues $\mathbf{\Lambda}$ and eigenvectors $\mathbf{R}$; finally, the eigenvectors of the initial problem are calculated by $\mathbf{\Phi} = \mathbf{YR}$.

Let us concatenate column-wise into a matrix $\mathbf{H}$ all the eigenvector matrices $\mathbf{\Phi}_i, i = 1, \ldots, M$, and the differences of all possible pairs of the means $\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j$, where $i, j = 1, \ldots, M$ and $j > i$ as in (2.6):

$$\mathbf{H} = \left[\mathbf{\Phi}_1 | \mathbf{\Phi}_2 | \ldots | \mathbf{\Phi}_M | \left(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2\right) | \left(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_3\right) | \ldots | \left(\bar{\mathbf{x}}_{M-1} - \bar{\mathbf{x}}_M\right)\right] \tag{2.7}$$

By orthonormalizing $\mathbf{H}$ the $n \times p$ basis $\mathbf{Y}$ is obtained

$$\mathbf{Y} = Orth\left(\mathbf{H}\right) \tag{2.8}$$

where p is determined by any orthonormalization algorithm.

Now, consider the following intermediate problem

$$\mathbf{D} = \mathbf{Y}\mathbf{R}\mathbf{\Lambda}\mathbf{R}^T\mathbf{Y}^T \tag{2.9}$$

where $\mathbf{Y}$ is the basis and $\mathbf{R}$ can be considered as a rotation matrix [20]. Substituting here the expression for $\mathbf{D}$ we obtain

$$\begin{aligned}
&\mathbf{Y}^T\mathbf{D}\mathbf{Y} \triangleq \mathbf{R}\mathbf{\Lambda}\mathbf{R}^T \\
&\mathbf{Y}^T\mathbf{D}\mathbf{Y} = \frac{1}{1-\sum_{i=1}^{M} N_i p_i^2}\mathbf{Y}^T\left[\sum_{i=1}^{M}\left(N_i-1\right)\mathbf{D}_i p_i + \right. \\
&\left. + \sum_{i=1}^{M}\sum_{j=i+1}^{M} N_i N_j p_i p_j \left(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\right)\left(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\right)^T\right]\mathbf{Y} = \\
&= \frac{1}{1-\sum_{i=1}^{M} N_i p_i^2}\left\{\sum_{i=1}^{M}\left(N_i-1\right)\left(\mathbf{Y}^T\mathbf{\Phi}_i\right)\mathbf{\Lambda}_i\left(\mathbf{Y}^T\mathbf{\Phi}_i\right)^T p_i + \right. \\
&\left. + \sum_{i=1}^{M}\sum_{j=i+1}^{M} N_i N_j p_i p_j \left(\mathbf{Y}^T\left[\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\right]\right)\left(\mathbf{Y}^T\left[\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\right]\right)^T\right\}
\end{aligned}$$

Note that it is advantageous to calculate $\mathbf{Y}^T\mathbf{\Phi}_i$ and $\mathbf{Y}^T\left[\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\right]$ first, because $\mathbf{Y}$ and $\mathbf{\Phi}_i$ have fewer columns than rows. On the other hand $\mathbf{Y}^T\mathbf{D}\mathbf{Y}$ is a $p \times p$ matrix, which compared to the $n \times n$ matrix $\mathbf{D}$, in general, is smaller and therefore its eigendecomposition will be faster to perform. Now, using eigendecomposition, $\mathbf{R}$ and $\mathbf{\Lambda}$ can easily be calculated. The resulting eigenvectors are obtained by

$$\mathbf{\Phi} = \mathbf{Y}\mathbf{R} \tag{2.10}$$

whereupon zero and small eigenvalues and the corresponding eigenvectors can be discarded to further reduce dimensionality.

This concludes the section on eigenspace fusion. Fig. 2.1 illustrates the fusion of three eigenspaces, represented by hyper-ellipses, with equal weights. The largest hyper-ellipse is the fused eigenspace and the small circles inside the hyper-ellipses are the observations. The following sections apply the above concepts to develop a framework for ASM and AAM fusion.

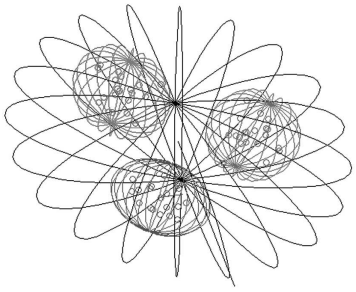


Figure 2.1: Illustration of eigenspace fusion. The three smallest ellipsoids are the fused eigenspaces and the largest one is the result of their fusion with equal weights. Small circles inside the ellipsoids are the original observations.

2.3 ASM Fusion

An Active Shape Model is constructed from a set of aligned shapes by means of PCA [7]. Shapes are defined by landmark points placed along the contour of the object of interest. ASM's main component is a Point Distribution Model (PDM) defined for the i-th ASM by:

$$\mathbf{x}_{ij} = \bar{\mathbf{x}}_i + \mathbf{\Phi}_i \mathbf{b}_{ij}^s \tag{2.11}$$

where $\mathbf{x}_{ij}$ is a n-vector, representing the j-th shape, obtained by concatenating all the landmark coordinates into a single real-valued vector one after another. In other words, if landmarks have coordinates (x_i, y_i) the concatenated vector will be of the form $(x_1, y_1, x_2, y_2, \ldots)^T$. Then, the n-vector $\bar{\mathbf{x}}_i$ is the mean of the aligned shapes in the training set; the $n \times m_i$ matrix $\mathbf{\Phi}_i$ and the m_i-vector $\mathbf{b}_{ij}^s$ are the projection matrix and the corresponding projection coordinates, respectively. Values n and m_i have been determined during the construction of PDMs to be fused.

To fit the model to an image, profiles perpendicular to the contour at landmark positions are used. From pixels sampled along each profile the mean vector and covariance matrix are estimated during the model construction. The collection of such pairs for each landmark constitutes an *Intensity Model*. They are a part of the Mahalanobis distance, which is used to drive the model to the best-fit location during segmentation.

ASM fusion is straightforward. Since each PDM is nothing else than an eigenspace containing the shapes from the training set, the fusion of several ASMs is reduced to the fusion of "aligned" PDM eigenspaces (using the algorithm from Section 2.2) and fusion of statistical information for the shape profiles.

The first step is to align the means $\bar{\mathbf{x}}_i$ of all PDM's using the Procrustes Analysis with the only difference being that, instead of the usual mean, the weighted mean is estimated according to (2.3). After Procrustes Analysis the aligned means $\bar{\mathbf{x}}_i$ are used to estimate the fused mean $\bar{\mathbf{x}}$ as in (2.3). During the alignment, the shapes are centered and rescaled to unit size and then a $d \times d$ matrix accounting for rotation is estimated, where d is the dimensionality of landmarks (*e.g.*, if landmarks represent

a shape in 2D then $d = 2$, if in 3D then $d = 3$). Let $\mathbf{S}_i$ be this $d \times d$ matrix that aligns the shape $\bar{\mathbf{x}}_i$ to the mean $\bar{\mathbf{x}}$, both being centered at the origin. Let $\boldsymbol{\Xi}_i$ be a $\frac{n}{d} \times \frac{n}{d}$ block-diagonal matrix with repeating $\mathbf{S}_i$ along its diagonal:

$$\boldsymbol{\Xi}_i = \begin{bmatrix} \mathbf{S}_i & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{S}_i \end{bmatrix} \tag{2.12}$$

Then, considering that the shapes are already centered, we can write

$$\boldsymbol{\Xi}_i \mathbf{x}_{ij} = \boldsymbol{\Xi}_i \bar{\mathbf{x}}_i + \boldsymbol{\Xi}_i \boldsymbol{\Phi}_i \mathbf{b}^s_{ij} \tag{2.13}$$

which shows that by aligning $\bar{\mathbf{x}}_i$ and each eigenvector of $\boldsymbol{\Phi}_i$ we also align the original observations $\mathbf{x}_{ij}$ to the fused mean $\bar{\mathbf{x}}$.

Now, using the transformations $\boldsymbol{\Xi}_i$, PDMs are fused by applying the eigenspace fusion scheme to eigenspaces corresponding to the aligned means $\boldsymbol{\Xi}_i \bar{\mathbf{x}}_i$ and eigenvectors $\boldsymbol{\Xi}_i \boldsymbol{\Phi}_i$.

The fusion of Intensity Models of ASMs is much simpler to perform, for it does not require the construction of the intermediate problem. The mean profiles of each model at each landmark are fused using (2.3) and the covariance matrices using (2.6), which directly uses covariance matrices instead of their eigendecompositions.

One may notice that the center of the fused eigenspace in Fig. 2.1 incidentally does not belong to any of three clusters of points (in this specific case) and one may suspect that the fused PDM will thus represent implausible shapes. But it must be noted that the fusion framework was developed in such a way that fusing two eigenspaces, constructed from several sets of data, is equivalent to constructing a new eigenspace from all of the sets. Therefore if constructing the model from all the observations is meaningful, then the fusion will result in a meaningful model too. Hence the apparent inconsistency reveals rather a feature of data than of the fusion technique.

2.4 AAM Fusion

2.4.1 Introduction

Let us assume that we are given M AAMs and that each model has an associated weight w_i such that they altogether add to one. The first component of an AAM that we are going to consider is a PDM, which describes the shape variation (learnt from training set) of the object of interest. The classical linear PDM used in AAM is defined as in the previous section by:

$$\mathbf{x}_{ij} = \bar{\mathbf{x}}_i + \boldsymbol{\Phi}_{si} \mathbf{b}^s_{ij}, \quad i = 1, \ldots, M \tag{2.14}$$

where the n-vector $\mathbf{x}_{ij}$ is the j-th training shape instance for the i-th PDM, the n-vector $\bar{\mathbf{x}}_i$ is the mean shape for i-th PDM, the $n \times m_i$ matrix $\mathbf{\Phi}_{si}$ is the matrix of eigenvectors, and $\mathbf{b}_{ij}^s$ are m_i coordinates of $\mathbf{x}_{ij}$ in the subspace spanned by $\mathbf{\Phi}_{si}$. The eigenspace associated with each PDM is $\Omega_{si} = (\bar{\mathbf{x}}_i, \mathbf{\Phi}_{si}, \mathbf{\Lambda}_{si}, N_i)$. The letter "$s$" in subscript or superscript of a symbol relates the latter to the PDM. It must be also mentioned that our technique requires that all the PDMs use the same landmark placement and, thus, to have the same number of landmarks.

The next component of an AAM is a Texture Model (TM). TMs are constructed from the intensity values of pixels inside the shape.

A linear TM is defined by

$$\mathbf{g}_{ij} = \bar{\mathbf{g}}_i + \mathbf{\Phi}_{gi}\mathbf{b}_{ij}^g, \quad i = 1, \ldots, M \tag{2.15}$$

where the k_i-vector $\mathbf{g}_{ij}$ is the j-th texture instance for the i-th TM, the k_i-vector $\bar{\mathbf{g}}_i$ is the mean texture for i-th model, the $k_i \times l_i$ matrix $\mathbf{\Phi}_{gi}$ is the matrix of eigenvectors, and $\mathbf{b}_{ij}^g$ are the l_i projections of $\mathbf{g}_{ij}$ in the subspace spanned by $\mathbf{\Phi}_{gi}$. The corresponding eigenspaces are $\Omega_{gi} = \left(\bar{\mathbf{g}}_i, \mathbf{\Phi}_{gi}, \mathbf{\Lambda}_{gi}, N_i\right)$. The values k_i and l_i have been determined during the construction of the TMs to be fused. The letter "g" in subscript or superscript of a symbol relates the symbol to the TM.

Having parameterized shape and texture, a combined AAM is constructed and defined by:

$$\mathbf{b}_{ij} = \mathbf{\Phi}_{ci}\, \mathbf{c}_{ij}, \quad i = 1, \ldots, M; \quad j = 1, \ldots, N_i \tag{2.16}$$

with a $(m_i + l_i) \times q_i$ eigenvector matrix $\mathbf{\Phi}_{ci}$. The $(m_i + l_i)$-vector $\mathbf{b}_{ij}$ is constructed from the corresponding i-th TM parameters and i-th PDM parameters as follows:

$$\mathbf{b}_{ij} = \begin{bmatrix} \mathbf{W}_{ci} \cdot \mathbf{b}_{ij}^s \\ \mathbf{b}_{ij}^g \end{bmatrix} = \widetilde{\mathbf{W}}_{ci} \cdot \begin{bmatrix} \mathbf{b}_{ij}^s \\ \mathbf{b}_{ij}^g \end{bmatrix}; \quad \widetilde{\mathbf{W}}_{ci} = \begin{bmatrix} \mathbf{W}_{ci} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{2.17}$$

$\mathbf{W}_{ci}$ being a diagonal $m_i \times m_i$ matrix of weights, calculated from the eigenvalues of the PDM and TM, used to make shape and texture parameters commeasurable [8]. The corresponding eigenspaces are defined by $\Omega_{ci} = (\mathbf{0}, \mathbf{\Phi}_{ci}, \mathbf{\Lambda}_{ci}, N_i)$. The value q_i has been determined during construction of the i-th ASM's combined model. The letter "c" in subscript or superscript of a symbol relates the symbol to the combined model.

Matching the above combined AAM to an image is performed in an iterative manner using the prediction matrices calculated during model construction. These matrices provide a linear relationship between the differences in texture and the differences in parameters of the corresponding model. Each of these matrices being multiplied by the difference between the sampled and modeled texture gives as a result the difference of the model parameters. The latter provides updates to the

model driving it to the best fit by minimizing the texture difference [30]. In this context, the following equalities hold:

$$\begin{aligned} \mathbf{R}_c \cdot \mathbf{r}\left(\mathbf{c}^*, \mathbf{t}^*\right) &= \Delta\mathbf{c} \\ \mathbf{R}_t \cdot \mathbf{r}\left(\mathbf{c}^*, \mathbf{t}^*\right) &= \Delta\mathbf{t} \end{aligned} \tag{2.18}$$

where $\Delta\mathbf{c} = \mathbf{c} - \mathbf{c}^*$ and $\Delta\mathbf{t} = \mathbf{t} - \mathbf{t}^*$ are the differences in model and pose parameters, $\mathbf{c}^*$ and $\mathbf{t}^*$ are current estimates of the parameters and $\mathbf{r}\left(\mathbf{c}^*, \mathbf{t}^*\right) = \mathbf{g}_{image}\left(\mathbf{c}^*, \mathbf{t}^*\right) - \mathbf{g}_{model}\left(\mathbf{c}^*, \mathbf{t}^*\right)$ is the texture residual (*i.e.*, the difference between the texture sampled from the image under the shape generated by the model with parameters $\mathbf{c}^*$, $\mathbf{t}^*$ and the texture generated by the model with the same parameters). For a more detailed explanation of the AAM matching process, please refer to [30]. The matrices $\mathbf{R}_c$ and $\mathbf{R}_t$ are the inverse of the Jacobians that are calculated during model building, the former is used for estimating the displacement of model parameters and the latter for pose parameter displacement. Every j-th column of i-th Jacobian is calculated by the formula [30]:

$$\frac{\partial \mathbf{r}}{\partial q_j^i} = \frac{1}{N_i} \sum_{l=1}^{N_i} \sum_k K\left(\delta q_{jk}^i\right) \frac{\Delta \mathbf{r}_i^l}{2\delta q_{jk}^i} \tag{2.19}$$

where $K(\cdot)$ is a weighting kernel (*e.g.*, Gaussian) and $\Delta\mathbf{r}_i^l$ is the difference between residuals corresponding to positive and negative parameter displacements in the l-th image from the i-th training set [30] and δq_{jk}^i is the k-th displacement in parameter j for the i-th model.

Having considered the above information we come to the conclusion that fusing several Active Appearance Models involves the following steps:

1. Fusing the Point Distribution Models of different AAMs
2. Fusing the Texture Models of the AAMs
3. Creating a Combined Appearance Model from the fused Point Distribution and Texture models.
4. Combining the prediction matrices of AAMs

2.4.2 Fusing the Point Distribution Models

The PDMs are fused in exactly the same way as those of ASM, described in Section 2.3. As the result a fused PDM is obtained:

$$\mathbf{x}_j = \bar{\mathbf{x}} + \mathbf{\Phi}_s \mathbf{b}_j^s, \quad j = 1, \ldots, N \tag{2.20}$$

with the corresponding eigenspace $\Omega_s = (\bar{\mathbf{x}}, \mathbf{\Phi}_s, \mathbf{\Lambda}_s, N)$, $\mathbf{\Phi}_s$ being a $n \times m$ matrix with m determined during fusion.

To simplify notation, in subsequent formulation we will assume that the eigenvectors $\mathbf{\Phi}_{si}$ (but not the means) are already transformed by Ξ_i.

2.4.3 Fusing Texture Models

The fusion of Texture Models is more complex because the model deals with textures in vector form. Therefore all the information about spatial relationships among the pixels is lost and the texture vector is bound to the shape from which it was sampled (*i.e.,* to the mean shape of the model $\bar{\mathbf{x}}_i$, for some i). As a consequence, to fuse TMs, all the textures must be warped onto the fused mean shape $\bar{\mathbf{x}}$.

Let us define M warping functions of texture: the i-th warp corresponds to a mapping of the k_i-dimensional texture vector from the mean shape $\bar{\mathbf{x}}_i$ of the i-th PDM (2.14) to the k-dimensional texture vector corresponding to the mean shape $\bar{\mathbf{x}}$ of the fused PDM (2.20):

$$\tau_i\left(\mathbf{g}_{ij}\right) = \widetilde{\mathbf{g}}_{ij} \tag{2.21}$$

where k is determined by the the number of pixels within the $\bar{\mathbf{x}}$. Note that the $\bar{\mathbf{x}}_i$ are not those aligned to $\bar{\mathbf{x}}$ during PDM fusion, but are the original ones.

The transformations (2.21) are linear functions of texture. This statement follows from the fact that these warp transformations only move pixels from one place to another with the help of interpolation when the required intensity does not have integer coordinates within the image. But any interpolation that is a linear function of pixel intensities, preserves the linearity of the warp (see Appendix II). Although linear interpolation is almost ubiquitous, this comment was made just in case someone would try to use the nonlinear one.

To fuse the Texture Models (2.15) all the textures have to be warped from their original shapes to the mean shape $\bar{\mathbf{x}}$. Due to the linearity of the warp

$$\tau_i\left(\mathbf{g}_{ij}\right) = \tau_i\left(\bar{\mathbf{g}}_i\right) + \tau_i\left(\mathbf{\Phi}_{gi}\right) \cdot \mathbf{b}_{ij}^{g} \tag{2.22}$$

where $\tau_i\left(\mathbf{\Phi}_{gi}\right)$ is a matrix whose columns are the columns of $\mathbf{\Phi}_{gi}$ warped by τ_i. Therefore warping $\mathbf{g}_{ij}$ is equivalent to warping the mean and the basis vectors, and the fused texture model is thus obtained by fusing the modified eigenspaces $\widetilde{\Omega}_{gi} = \left(\tau_i\left(\bar{\mathbf{g}}_i\right), \tau_i\left(\mathbf{\Phi}_{gi}\right), \mathbf{\Lambda}_{gi}, N_i\right), i = 1, \ldots, M$. One may note that $\tau_i\left(\mathbf{\Phi}_{gi}\right)$ is, in general, no longer orthonormal, but this is not required for fusion (see Appendix III).

Fusing all the $\widetilde{\Omega}_{gi}$ yields a fused TM corresponding to the Point Distribution Model (2.20)

$$\mathbf{g}_j = \bar{\mathbf{g}} + \mathbf{\Phi}_g \mathbf{b}_j^{g}, \quad j = 1, \ldots, N \tag{2.23}$$

with the eigenspace $\Omega_g = \left(\bar{\mathbf{g}}, \mathbf{\Phi}_g, \mathbf{\Lambda}_g, N\right)$, $\bar{\mathbf{g}}$ being a k-vector and $\mathbf{\Phi}_g$ a $k \times l$ matrix (l is determined during fusion).

2.4.4 Creating a Fused AAM

At this point, a fused AAM model has to be constructed from the fused PDM and TD models:

$$\mathbf{b}_j = \mathbf{\Phi}_c \, \mathbf{c}_j, \quad j = 1, \ldots, N \tag{2.24}$$

where $\mathbf{\Phi}_c$ is a $(m+l) \times q$ matrix (q is determined during fusion) and $\mathbf{b}_j$ is the following $(m+l)$-vector:

$$\mathbf{b}_j = \begin{bmatrix} \mathbf{W}_c \cdot \mathbf{b}_j^s \\ \mathbf{b}_j^g \end{bmatrix} = \widetilde{\mathbf{W}}_c \cdot \begin{bmatrix} \mathbf{b}_j^s \\ \mathbf{b}_j^g \end{bmatrix}; \quad \widetilde{\mathbf{W}}_c = \begin{bmatrix} \mathbf{W}_c & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$$

The matrix $\mathbf{W}_c$ is readily calculated from $\mathbf{\Lambda}_g$ and $\mathbf{\Lambda}_s$ as

$$\mathbf{W}_c = \mathbf{I} \cdot \sqrt{\frac{tr(\mathbf{\Lambda}_g)}{tr(\mathbf{\Lambda}_s)}}$$

$tr(\cdot)$ standing for the trace of a matrix.

By construction, the mean combined vector zero and, therefore, only $\mathbf{\Phi}_c$ and $\mathbf{\Lambda}_c$ remain to be estimated.

Let $\mathbf{C}_i = \mathbf{\Phi}_{ci}\mathbf{\Lambda}_{ci}\mathbf{\Phi}_{ci}^T$ be the covariance matrices of the corresponding AAMs' combined models (2.16), and let $\mathbf{C} = \mathbf{\Phi}_c\mathbf{\Lambda}_c\mathbf{\Phi}_c^T$ be the covariance matrix of the fused combined model (2.24).

It is possible to show that the relationships between the bases $\mathbf{\Phi}_{si}$ and $\mathbf{\Phi}_s$ are given by (see Appendix IV):

$$\mathbf{b}_j^s = \mathbf{\Phi}_s^{-1}\mathbf{\Phi}_{si}\mathbf{b}_{ij}^s \tag{2.25}$$

and similarly, the relationships between the bases $\mathbf{\Phi}_{gi}$ and $\mathbf{\Phi}_g$ are given by:

$$\mathbf{b}_j^g = \mathbf{T}_i \cdot \mathbf{b}_{ij}^g, \quad \mathbf{T}_i = \mathbf{\Phi}_g^{-1} \cdot \tau_i\left(\mathbf{\Phi}_{gi}\right) \tag{2.26}$$

with $\mathbf{T}_i$ being a $l \times l_i$ matrix.

Rewriting the expression for $\mathbf{C}_i$ (according to the definition of covariance matrix) we can obtain

$$\begin{aligned} (N_i - 1) \cdot \mathbf{C}_i &= (N_i - 1) \cdot \mathbf{\Phi}_{ci}\mathbf{\Lambda}_{ci}\mathbf{\Phi}_{ci}^T = \\ &= \widetilde{\mathbf{W}}_{ci} \cdot \left(\sum_{j=1}^{N_i} \begin{bmatrix} \mathbf{b}_{ij}^s \cdot \left(\mathbf{b}_{ij}^s\right)^T & \mathbf{b}_{ij}^s \cdot \left(\mathbf{b}_{ij}^g\right)^T \\ \mathbf{b}_{ij}^g \cdot \left(\mathbf{b}_{ij}^s\right)^T & \mathbf{b}_{ij}^g \cdot \left(\mathbf{b}_{ij}^g\right)^T \end{bmatrix} \right) \cdot \widetilde{\mathbf{W}}_{ci}^T \end{aligned}$$

or, equivalently,

$$\begin{aligned} \sum_{j=1}^{N_i} \begin{bmatrix} \mathbf{b}_{ij}^s \cdot \left(\mathbf{b}_{ij}^s\right)^T & \mathbf{b}_{ij}^s \cdot \left(\mathbf{b}_{ij}^g\right)^T \\ \mathbf{b}_{ij}^g \cdot \left(\mathbf{b}_{ij}^s\right)^T & \mathbf{b}_{ij}^g \cdot \left(\mathbf{b}_{ij}^g\right)^T \end{bmatrix} = \\ = (N_i - 1) \cdot \widetilde{\mathbf{W}}_{ci}^{-1} \cdot \mathbf{C}_i \cdot \widetilde{\mathbf{W}}_{ci}^{-T} \end{aligned} \tag{2.27}$$

Let us write the expression for the unbiased estimate of $\mathbf{C}$ (see Appendix I):

$$\left(1-\sum_{i=1}^{M} N_i p_i^2\right)\cdot\mathbf{C}= \\ =\widetilde{\mathbf{W}}_c\cdot\left(\sum_{j=1}^{N}\begin{bmatrix}\mathbf{b}_j^s\cdot\left(\mathbf{b}_j^s\right)^T & \mathbf{b}_j^s\cdot\left(\mathbf{b}_j^g\right)^T\\ \mathbf{b}_j^g\cdot\left(\mathbf{b}_j^s\right)^T & \mathbf{b}_j^g\cdot\left(\mathbf{b}_j^g\right)^T\end{bmatrix}\cdot\Pr\left(\mathbf{z}_j\right)\right)\cdot\widetilde{\mathbf{W}}_c^T$$

substituting the expressions (2.25), (2.26) into the last equation and using (2.27), it is easy to obtain the following

$$\mathbf{C}=\frac{1}{1-\sum_{i=1}^{M} N_i p_i^2}\cdot\mathbf{\Psi}\left(\sum_{i=1}^{M}\left\{\left(N_i-1\right)\cdot\mathbf{\Gamma}_i\mathbf{C}_i\mathbf{\Gamma}_i^T\cdot p_i\right\}\right)\mathbf{\Psi}^T \tag{2.28}$$

where

$$\mathbf{\Psi}=\begin{bmatrix}\mathbf{W}_c\mathbf{\Phi}_s^{-1} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\end{bmatrix};\quad \mathbf{\Gamma}_i=\begin{bmatrix}\mathbf{\Phi}_{si}\mathbf{W}_{ci}^{-1} & \mathbf{0}\\ \mathbf{0} & \mathbf{T}_i\end{bmatrix}$$

$\mathbf{\Psi}$ being a $(m+l)\times(n+l)$ matrix (the contained identity matrix $\mathbf{I}$ is a $l\times l$ matrix), and $\mathbf{\Gamma}_i$ being a $(n+l)\times(m_i+l_i)$ matrix.

$\mathbf{\Phi}_c$ and $\mathbf{\Lambda}_c$ are finally calculated by eigendecomposition of $\mathbf{C}$:

$$\mathbf{C}\overset{\Delta}{=}\mathbf{\Phi}_c\mathbf{\Lambda}_c\mathbf{\Phi}_c^T \tag{2.29}$$

The eigenproblem (2.29) can also be rewritten in the following form

$$\frac{\sum_{i=1}^{M}\left\{\left(N_i-1\right)p_i\cdot\left(\mathbf{\Psi}\mathbf{\Gamma}_i\mathbf{\Phi}_{ci}\right)\mathbf{\Lambda}_{ci}\left(\mathbf{\Psi}\mathbf{\Gamma}_i\mathbf{\Phi}_{ci}\right)^T\right\}}{1-\sum_{i=1}^{M} N_i p_i^2}\overset{\Delta}{=}\mathbf{\Phi}_c\mathbf{\Lambda}_c\mathbf{\Phi}_c^T$$

and solved using the eigenspace fusion algorithm from Section 2.2. To be more precise, the algorithm should be applied to the eigenspaces $\widetilde{\Omega}_{ci}=\left(\mathbf{0},\mathbf{\Psi}\mathbf{\Gamma}_i\mathbf{\Phi}_{ci},\mathbf{\Lambda}_{ci},N_i\right)$.

2.5 Fusing the Prediction Matrices

Due to (2.19), the j-th column for the fused Jacobian is as follows:

$$\frac{\partial\mathbf{r}}{\partial q_j}=\sum_{i=1}^{M}p_i\left[\sum_{l=1}^{N_i}\sum_{k}K\left(\delta q_{jk}^i\right)\frac{\Delta\mathbf{r}_i^l}{2\delta q_{jk}^i}\right]=\sum_{i=1}^{M}N_i p_i\left[\frac{\partial\mathbf{r}}{\partial q_j^i}\right]$$

or denoting the M model Jacobians by $\mathbf{J}_{ci} = \mathbf{R}_{ci}^{-1}$ (each being a $k_i \times q_i$ matrix), the pose Jacobians by $\mathbf{J}_{ti} = \mathbf{R}_{ti}^{-1}$ ($k_i \times 2d$ matrix each[1], where d is the dimensionality of the landmarks), for $i = 1, \ldots, M$ ($\mathbf{R}_{ti}$ and $\mathbf{R}_{ci}$ are stored with the AAMs), and the fused model and pose Jacobians by $\mathbf{J}_c$ and $\mathbf{J}_t$, respectively, we can write

$$\mathbf{J}_c = \sum_{i=1}^{M} N_i p_i \mathbf{J}_{ci}, \quad \mathbf{J}_t = \sum_{i=1}^{M} N_i p_i \mathbf{J}_{ti} \tag{2.30}$$

There are several issues that prevent direct application of the formula (2.30): the Jacobians, in general, have different number of rows and columns. In other words, different AAMs have different number of parameters and different texture lengths. First, let us deal with the differences in the texture length. As we already did for texture models, we must warp the Jacobians. So we warp each column of each Jacobian (using τ_i for $\mathbf{J}_i$) to obtain the new ones, such that they are $k \times q$ and $k \times 2d$ matrices

$$\widetilde{\mathbf{J}}_{ci} = \tau_i \left(\mathbf{J}_{ci} \right), \quad \widetilde{\mathbf{J}}_{ti} = \tau_i \left(\mathbf{J}_{ti} \right)$$

Now we need to handle the problem of different number of parameters. Since there is a constant number of parameters for the pose, $\widetilde{\mathbf{J}}_{ti}$ requires no special handling.

Let us rewrite the part of (2.18) responsible for i-th model parameters

$$\widetilde{\mathbf{J}}_{ci} \cdot \Delta \mathbf{c}_{ij} = \mathbf{r} \left(\mathbf{c}^*, \mathbf{t}^* \right) \tag{2.31}$$

By substituting the coordinate transformations (2.25) and (2.26) into the definition of the combined model (2.16), we can obtain the transformation between $\mathbf{c}_{ij}$ and $\mathbf{c}_j$

$$\begin{bmatrix} \mathbf{W}_{ci} \mathbf{b}_{ij}^s \\ \mathbf{b}_{ij}^g \end{bmatrix} = \mathbf{\Phi}_{ci} \mathbf{c}_{ij} \Rightarrow \begin{bmatrix} \mathbf{W}_{ci} \mathbf{\Phi}_{si}^{-1} \mathbf{\Phi}_s \mathbf{b}_j^s \\ \mathbf{T}_i^{-1} \mathbf{b}_j^g \end{bmatrix} = \mathbf{\Phi}_{ci} \mathbf{c}_{ij}$$

$$\begin{pmatrix} \mathbf{W}_{ci} \mathbf{\Phi}_{si}^{-1} \mathbf{\Phi}_s \mathbf{W}_c^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_i^{-1} \end{pmatrix} \begin{bmatrix} \mathbf{W}_c \mathbf{b}_j^s \\ \mathbf{b}_j^g \end{bmatrix} = \mathbf{\Phi}_{ci} \mathbf{c}_{ij}$$

$$\widetilde{\mathbf{\Phi}}_{ci} = \begin{pmatrix} \mathbf{W}_{ci} \mathbf{\Phi}_{si}^{-1} \mathbf{\Phi}_s \mathbf{W}_c^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_i^{-1} \end{pmatrix}^{-1} \cdot \mathbf{\Phi}_{ci} = \mathbf{\Psi} \mathbf{\Gamma}_i \mathbf{\Phi}_{ci}$$

$$\begin{bmatrix} \mathbf{W}_c \mathbf{b}_j^s \\ \mathbf{b}_j^g \end{bmatrix} = \widetilde{\mathbf{\Phi}}_{ci} \mathbf{c}_{ij}$$

The transformation is as follows

$$\mathbf{c}_{ij} = \widetilde{\mathbf{\Phi}}_{ci}^{-1} \cdot \mathbf{\Phi}_c \cdot \mathbf{c}_j \tag{2.32}$$

[1] d translation parameters, $d - 1$ angles and one scaling parameter

Substituting (2.32) into (2.31) we obtain

$$\tilde{\mathbf{J}}_{ci}\tilde{\mathbf{\Phi}}_{ci}^{-1}\mathbf{\Phi}_c\Delta\mathbf{c}_j = \mathbf{r}\left(\mathbf{c}^*, \mathbf{t}^*\right)$$

Summarizing, we can rewrite (2.18) as

$$\begin{aligned}\left[\tau_i\left(\mathbf{J}_{ci}\right)\cdot\widetilde{\mathbf{\Phi}}_{ci}^{-1}\mathbf{\Phi}_c\right]\cdot\Delta\mathbf{c}_j &= \mathbf{r}\left(\mathbf{c}^*, \mathbf{t}^*\right)\\ \tau_i\left(\mathbf{J}_{ti}\right)\cdot\Delta\mathbf{t}_j &= \mathbf{r}\left(\mathbf{c}^*, \mathbf{t}^*\right)\end{aligned}$$

providing a transformation for each Jacobian:

$$\hat{\mathbf{J}}_{ci} = \tau_i\left(\mathbf{J}_{ci}\right)\widetilde{\mathbf{\Phi}}_{ci}^{-1}\mathbf{\Phi}_c, \quad \hat{\mathbf{J}}_{ti} = \tau_i\left(\mathbf{J}_{ti}\right)$$

and finally the fused Jacobians are

$$\mathbf{J}_c = \sum_{i=1}^{M} N_i p_i \hat{\mathbf{J}}_{ci}, \quad \mathbf{J}_t = \sum_{i=1}^{M} N_i p_i \hat{\mathbf{J}}_{ti}$$

The fused prediction matrices are calculated by

$$\begin{aligned}\mathbf{R}_c = \mathbf{J}_c^{-1} &= \left[\left(\sum_{i=1}^{M} N_i p_i \cdot \tau_i\left(\mathbf{R}_{ci}^{-1}\right)\widetilde{\mathbf{\Phi}}_{ci}^{-1}\right)\cdot\mathbf{\Phi}_c\right]^{-1}\\ \mathbf{R}_t = \mathbf{J}_t^{-1} &= \left[\sum_{i=1}^{M} N_i p_i \cdot \tau_i\left(\mathbf{R}_{ti}^{-1}\right)\right]^{-1}\end{aligned} \tag{2.33}$$

being $q \times k$ and $2d \times k$ matrices respectively.

The obtained prediction matrices are ready to be plugged into the usual AAM matching algorithm [30] to link pixel intensity differences with the displacements of model parameters.

2.6 AAM Fusion Algorithm Outline

Let us briefly summarize here the steps required to fuse Active Appearance Models

1. *PDM Fusion.* Align the mean shapes $\bar{\mathbf{x}}_i$ by Procrustes Analysis. Compute the fused mean shape $\bar{\mathbf{x}}$ and the transformations Ξ_i that align $\bar{\mathbf{x}}_i$ to $\bar{\mathbf{x}}$ for every i-th model. Use each of the Ξ_i to transform the eigenvectors of the corresponding PDM. Fuse the eigenspaces of the input PDMs with the modified means and the transformed eigenvectors by the proposed eigenspace fusion algorithm to obtain a fused PDM (2.20).

Figure 2.2: The four expressions taken from the AR database (from left to right): neutral, smiling, angry, and screaming.

2. *TM Fusion.* Calculate the spatial warps $\tau_i(\tilde{\mathbf{g}}_i)$ and $\tau_i(\mathbf{\Phi}_{gi})$ in order to update the TM eigenspaces to $\widetilde{\Omega}_{gi} = (\tau_i(\tilde{\mathbf{g}}_i), \tau_i(\mathbf{\Phi}_{gi}), \mathbf{\Lambda}_{gi}, N_i), i = 1, \ldots, M$. Fuse them using the proposed eigenspace fusion algorithm to obtain the fused TM (2.23).

3. *Fusion of the Combined Models.* Calculate $\mathbf{W}_c$ and fuse the eigenspaces $\widetilde{\Omega}_{ci} = (\mathbf{0}, \mathbf{\Psi\Gamma}_i\mathbf{\Phi}_{ci}, \mathbf{\Lambda}_{ci}, N_i)$ to obtain the eigenvalues and eigenvectors for the fused AAM (see Sec. 2.4.4). Remember that $\mathbf{\Phi}_{si}$ appearing in $\mathbf{\Gamma}_i$ are the re-aligned PDM's eigenvectors.

4. *Fusion of the Prediction Matrices.* Fuse the prediction matrices using (2.33).

2.7 Results

In this section we will use frequently such terms as *fused* and *full* model. Given a set of observations to use for model construction, by *fused* model we mean that the set was split into two subsets, a model was constructed from each subset, and those models were fused with equal weights. Correspondingly the *full* model is the model constructed directly from all the observations of the set at once.

To illustrate and evaluate the developed framework, the AR database [31] has been chosen. Four expressions of 133 men and women, as in Fig. 2.2, were taken for testing. All the faces had been landmarked using the 98-point template shown in Fig. 2.3.

Firstly, we compare the *fused* and *full* models on the basis of the compactness, generalization and specificity measures of the fused AAM (since the PDM is a part of the ASM, the tests will also demonstrate the equivalence of the *fused* and *full* ASMs). These measures, computed for different number of model modes m, are formally given by the following formulas [32]:

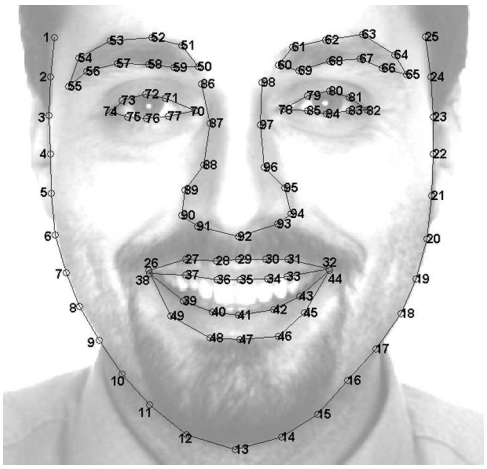


Figure 2.3: 98-point landmarking template used in experiments. Each landmark has a number associated with it for the reference.

- *Compactness* is defined by

$$\mathcal{C}(m) = \frac{1}{tr(\mathbf{\Lambda}_c)} \sum_{i=1}^{m} \lambda_i$$

where λ_i is i-th largest eigenvalue from $\mathbf{\Lambda}_c$ (2.29).

- *Generalization* is computed by

$$\mathcal{G}(m) = \frac{1}{|\mathbb{U}|} \sum_{i=1}^{|\mathbb{U}|} \|\hat{\mathbf{u}}_i(m) - \mathbf{u}_i\|$$

where $\|\cdot\|$ is the Euclidean or L^2 norm, $|\cdot|$ stands for the cardinality of a set, $\mathbb{U}$ is the training set of observations, $\hat{\mathbf{u}}_i(m)$ is an approximation to the observation $\mathbf{u}_i$ obtained by the model, constructed from the set $\mathbb{U} \setminus \{\mathbf{u}_i\}$, using only the m first modes of variation.

- *Specificity* is assessed, finally, by

$$\mathcal{S}(m) = \frac{1}{|\mathbb{V}|} \sum_{i=1}^{|\mathbb{V}|} \|\mathbf{v}_i(m) - \widetilde{\mathbf{u}}(\mathbf{v}_i)\|$$

where $\widetilde{\mathbf{u}}(\mathbf{v}_i) = \arg\min_{\mathbf{u} \in \mathbb{U}} \|\mathbf{v}_i(m) - \mathbf{u}\|$, and $\mathbb{V}$ is a set of observations corresponding to the random sampling of model's subspace (parameter space of PDM or TM defined by eigenvectors).

These measures were originally proposed for PDM evaluation, but we shall use them to evaluate the combined model of AAM (2.24). Since the latter describes

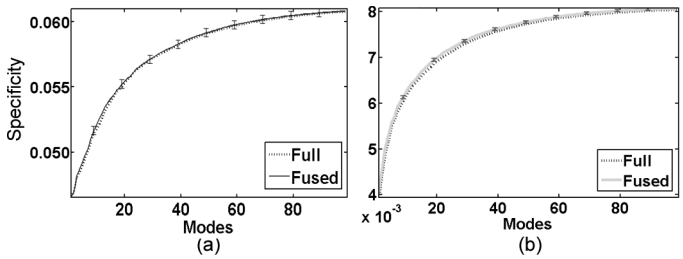


Figure 2.4: *Specificity* of the combined model of the *full* and *fused* AAMs: (a) computed from shape distances and normalized by the distance between the eyes; (b) computed from texture distances.

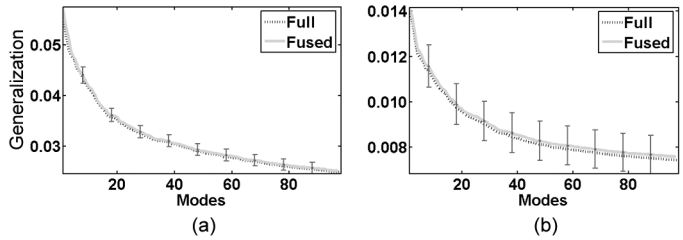


Figure 2.5: *Generalization* of the combined model of the *full* and *fused* AAMs: (a) computed from shape distances and normalized by the distance between the eyes; (b) computed from texture distances.

variation in both shape and texture, the *specificity* and *generalization* are computed separately in terms of shape and texture. To make the measurements in terms of shape independent of the face size, they are normalized by the distance between the two eye-centers in the manually landmarked face.

To evaluate the above mentioned performance measures the subjects from the AR database were randomly divided into two subsets of equal size. The *full* and the *fused* model were constructed from these sets, followed by calculating the above mentioned figures of merit. Figs. 2.4-2.6 provide the plots of *specificity*, *generalization* and *compactness*, respectively, of the combined model of the AAMs. The bars show 95% confidence interval (according to the t-Test) for the hypothesis that the performance of both *full* and *fused* models is equal.

To investigate the performance of the *fused* model in terms of segmentation accuracy, 10 segmentation experiments were performed, each consisting of the following steps. From the total of 532 AR database images, 266 (one half) were randomly chosen as a training set and randomly split into halves: training subset 1 and training subset 2. The whole training set of 266 images was used to construct the *full* ASM and AAM models. Training subsets 1 and 2 were used to construct two AAM and

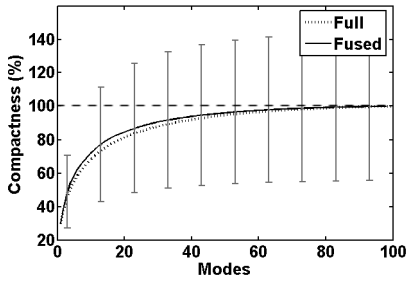


Figure 2.6: *Compactness* of the combined model of the *full* and *fused* AAMs.

two ASM submodels which were subsequently fused with equal weights to get the *fused* AAM and ASM. The remaining 266 images (not used in model construction) were used to test the segmentation performance of both fused AAM and ASM. Each image in this testing set was segmented by the *full* and the *fused* AAM and ASM and point-to-point errors were computed with respect to the manual segmentation. To provide a scale independent measurement, every error was normalized by the inter-eye distance (between the eye centers) of the corresponding manually landmarked shape. In other words the error is given as a percentage of the inter-eye distance. The mean point-to-point segmentation errors together with their 95% confidence intervals are presented in Fig. 2.7 for ASM tests and Fig. 2.8 for AAM tests. As it can be seen from these figures the fused ASM model behaves like a normal ASM model, while the fused AAM exhibits some very small difference of unlikely practical relevance.

The performance comparison in terms of speed can be found in Table 2.1. The first column shows the total number of observations used for model construction. To construct a fused model this set was split in halves, two submodels were constructed and the time it took to fuse them is displayed in the table in "AAM Fusion" and "ASM Fusion" columns. The total time of constructing the two submodels and fusing them is shown in the "AAM Fused" and "ASM Fused" columns. The "AAM Batch" and "ASM Batch" columns display the time of a batch construction of a corresponding model from all the observations. Note that the comparison was performed for 200, 400 and 600 images but the table also shows batch construction time for 100 and 300 images to show the time it takes to construct the submodels. It can be noted that for AAM there is a significant time saving when the model is constructed by the fusion of two submodels, while there is no difference for ASM. Nevertheless in a scenario when different weighting of observations is required (for example the continuous model update giving the new observations more weight) a complete batch model construction would be required at each update, while using the fusion, only one model from the new observations should be constructed and fused with the already constructed model of the old observations. And in this case the time saving is even bigger.

Table 2.1: Fusion Execution-Time Comparison[1]

#	AAM			ASM		
Images	**Batch**	**Fusion**	**Fused**	**Batch**	**Fusion**	**Fused**
100	2′ 02″	-	-	0′ 25″	-	-
200	7′ 25″	0′ 23″	4′ 27″	0′ 49″	0′ 1.0″	0′ 51″
300	12′ 38″	-	-	1′ 06″	-	-
400	21′ 30″	0′ 50″	15′ 40″	1′ 38″	0′ 1.2″	1′ 39″
600	41′ 50″	1′ 11″	26′ 27″	2′ 13″	0′ 1.3″	2′ 13″

[1] Evaluated on P4 2.80GHz, Intel D875PBZ Motherboard, 1Gb RAM. The time is given in " mm′ ss.s″ " format.

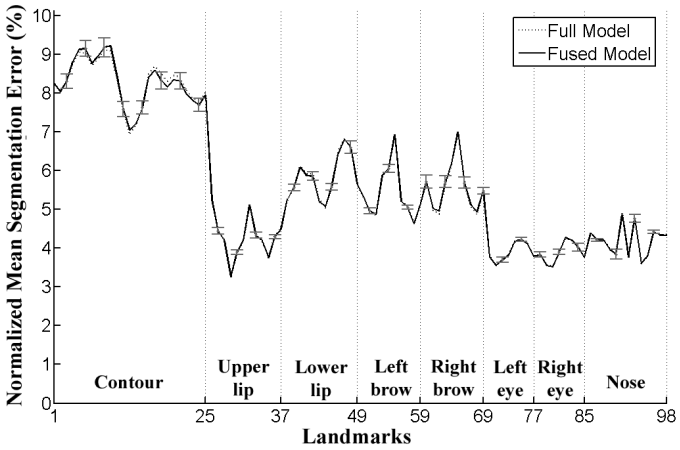


Figure 2.7: Comparison of the mean segmentation errors, as a percentage of the inter-eye distance, for ASM *fused* and *full* models in 10 experiments using the test sets from the AR database.

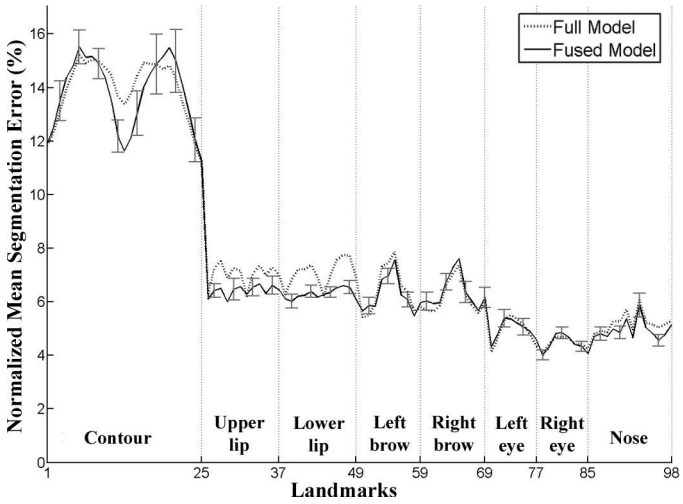


Figure 2.8: Comparison of the mean segmentation errors, as a percentage of the inter-eye distance, for AAM *fused* and *full* models in 10 experiments using the test sets from the AR database.

As a second illustration of our technique, we performed an identity verification test in order to analyze whether by fusing models constructed from different databases an increase in verification performance can be obtained. In this experiment two AAMs were constructed from two databases: AR [31] and Equinox [33]. Then these models were fused with equal weights to obtain a fused AAM. Finally the fused model together with the models constructed from each of the databases were used to segment images from a third database: XM2VTS [34]. Using the segmentation results, classification tests were performed on the XM2VTS database. To extract the features for classification, all the images were segmented by AAMs, and the texture was sampled from the resulting shapes. These textures were then projected onto the subspace of the texture model of the corresponding AAM. The resulting parameters were used for classification. The angular distance between vectors was taken as a distance measure [35]. DET curves for all three models, according to two standard configurations [34], are shown in Fig. 2.9. The curve corresponding to the fused model is, in principle, between the other two curves and there is no improvement. This is likely because both AR and Equinox databases have faces with the same expressions (except for widely open mouths in AR) and the fusion introduced no additional information. The model constructed from the Equinox database demonstrated the best performance because it has much less variation in expression while the AR has screaming faces, which introduce large variations into the data.

Finally, we demonstrate how fusion can improve segmentation by adding new

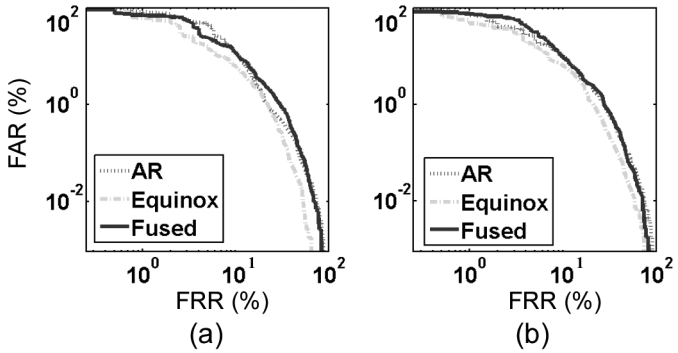


Figure 2.9: DET curves of classification tests on XM2VTS database using the fused model and the models built from the AR and EQUINOX databases. Classification is performed according to the configurations 1(a) and 2(b) [34], plotted on logarithmic scale.

information. To that end, two subsets were extracted from the database: faces with closed and open mouths (first and fourth expression in Fig. 2.2, 133 images each). Two AAM models were built from these sets, one from closed-mouth faces and one from open-mouth faces. Thence the model built from one set should not be able to segment any expression from the other one (*e.g.,* the model built from closed mouths, having no open mouths in the training set, cannot represent an open mouth). Then these models were fused with equal weights to form the *fused* AAM. To evaluate the performance, a segmentation of 133 faces with half-open mouth (second expression in Fig. 2.2) has been performed using both the original models and the fused one. Mean point-to-point segmentation errors, normalized by the inter-eye distance, together with 95% confidence intervals are shown in Fig. 2.10. An example of such a segmentation can be seen in Fig. 2.11. The *fused* model in Fig. 2.11a exhibits a significant improvement of mouth segmentation when compared to the AAMs constructed only from the closed-mouth or open-mouth images (Fig. 2.11bc). The fused AAM, having more information about the mouth variability, was able to segment it with much higher precision than the other two models.

2.8 Conclusions and Future Work

The article presented a method to fuse Active Shape and Active Appearance models, as well as a generalization of the eigenspace fusion algorithm originally proposed in [20]. Experiments demonstrate that, in practice, the fused ASM performs similarly to the full ASM, while the fused AAM slightly differs from the full AAM but with unlikely practical implications.

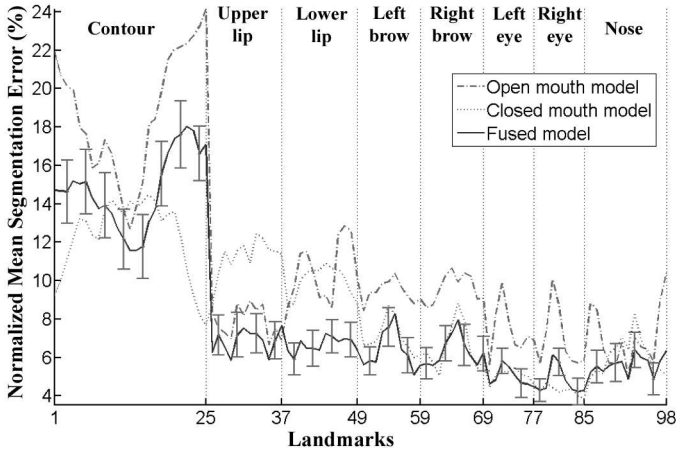


Figure 2.10: Comparison of the mean segmentation errors, as a percentage of the inter-eye distance, of the open-mouth model, closed-mouth model, and their fusion.

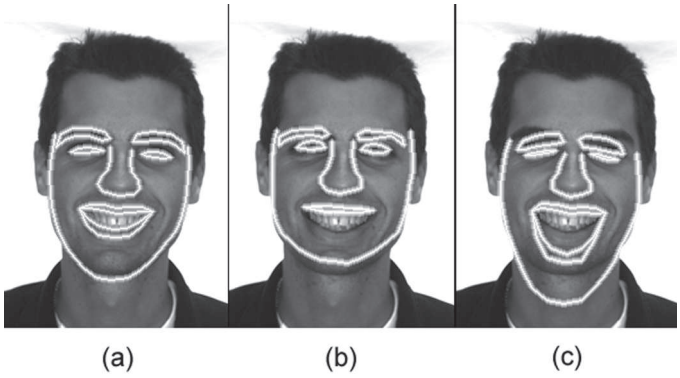


Figure 2.11: Example segmentations by the fused AAM (a) against closed-mouth (b) and open-mouth (c) models.

On the other hand, we can conclude that, essentially, the fusion is useful when either new information (such as new expressions in the last experiment of Section 2.7) can be introduced as a result of fusion, when the model needs to be updated online, or when the original observations are unavailable to reconstruct the model from both past and new observations. The proposed technique paves the way for saving time during AAM model construction by splitting the training set in several subsets and parallelizing the training procedure on each subset separately. Having constructed one AAM per subset, they can be fused using the proposed framework.

CHAPTER 3

Multi-View Face Segmentation Using Fusion of Statistical Shape and Appearance Models

Abstract - *This paper demonstrates how a weighted fusion of multiple Active Shape (ASM) or Active Appearance (AAM) models can be utilized to perform multi-view facial segmentation with only a limited number of views available for training the models. The idea is to construct models only from frontal and profile views and subsequently fuse these models with adequate weights to segment any facial view. This reduces the problem of multi-view facial segmentation to that of weight estimation, the algorithm for which is proposed as well. The evaluation is performed on a set of 280 landmarked static face images corresponding to seven different rotation angles and on several video sequences of the AV@CAR database. The evaluation demonstrates that the estimation of the weights does not have to be very accurate in the case of ASM, while in the case of AAM the influence of correct weight estimation is more critical. The segmentation with the proposed weight estimation method produced accurate segmentations in* 91% *of 280 testing images with the median point-to-point error varying from two to eight pixels (1.8%-7.2% of average inter-eye distance)*

Adapted from C. Butakoff, A.F. Frangi. Multi-View Face Segmentation Using Fusion of Statistical Shape and Appearance Models. *Computer Vision and Image Understanding*, in press, 2009.

3.1 Introduction

ESTIMATION or identification of the head's pose, or its separation from other facial information have attracted much research interest for quite some time. The 3D nature of head rotation poses a great challenge for any 2D face recognition or segmentation algorithm and, if not accounted for, can cause significant performance drops. Yet there are applications where it is important to be able to efficiently process different facial views. For instance, consider surveillance applications (*e.g.,* in an airport), where facial pose is usually impossible to be kept under control, or intelligent human-machine interfaces where head tracking can play an important role for interaction.

There are a number of strategies to handle multiple facial views, although most of them are applied to face recognition problems. For example, Gong *et al.* [36] represented faces by either normalized intensities or using the composite face representation scheme based on the Gabor wavelet transform. The authors investigated the possibility of identifying facial pose using the facial manifold. They have shown that images corresponding to pose changes of a continuous face rotation form a smooth curve in pose eigenspace. The authors also argue that it should be possible to construct a simple but generic face pose eigenspace, which can be used to estimate poses of unknown faces. The same idea was considered by Shih *et al.* [37], where multi-view face sequence is represented as a B-spline manifold. The Euclidean distance to the manifold is used to estimate the pose of the face in question. Another work that uses two-dimensional Gabor wavelet features for pose invariant face recognition is that by Gokberk *et al.* [38]. Support Vector Machines (SVM) were proposed for the problem of facial pose discrimination by Huang *et al.* [39]. Although, instead of estimating the head rotation angle, SVM is used to classify any given image as belonging to one of several available views (three views are considered: frontal, 33° rotation to the left and to the right). A similar approach was taken by Li *et al.* [40] but using a multi-class kernel support vector classifier instead of SVM and adding one extra class to represent non-faces. Another SVM-based pose estimation strategy can be found in Li *et al.* [41]. A pose differentiation by k-means clustering was proposed by Lee *et al.* [42]. Okada *et al.* [43, 44] proposed a model, coined PCMAP. It computes bidirectional mappings between facial images and physical parameters (3D head rotation angles), via parameterized manifold representations of faces in the PCA subspace. The model is subsequently used for view-independent face recognition. Finally, in a recent publication by Sanderson *et al.* [45], non-frontal views are artificially synthesized from the frontal ones using methods based on maximum likelihood linear regression and standard multi-variate linear regression.

Less work has been carried out in the area of pose-invariant 2D face segmentation. The most obvious approach to handle that problem is to train the segmentation algorithm with the data extracted from all the possible facial views, as for

example was done by Gonzalez-Jimenez *et al.* [46], and then use it to segment any face. Two other solutions were proposed by Cootes *et al.* [47]. The first one is to construct several models corresponding to different facial views. Subsequently, during segmentation, the one that best corresponds to the image is chosen. The second approach is to create a Coupled-View Appearance Model using PCA on pairs of opposite views, but this approach requires that the views represent exactly the same expression of the same face rotated by the same angle in opposite directions, in other words, ideally, they have to be captured simultaneously (the authors circumvented this difficulty using a mirror). The work by Gross *et al.* [48] proposed a modification of AAM to handle occlusions. The AAM was trained on faces with artificially generated occlusions. The head rotation was treated as partial occlusion of the face. Some work has been done in the area of ASMs as well. Wan *et al.* [49] proposed to decouple ASM of facial features from the ASM of facial contour and use genetic algorithm to match the model to an image. The method was evaluated on the ORL face database featuring left-right rotations of up to 45°. Buxton *et al.* [50] used projective geometry to adapt ASM to different viewpoints. Restricting the method to affine imaging conditions the pose variation is removed based on two reference views. Only the contours of facial features (without the contour of the face) are considered by the authors. In another work, Zhou *et al.* [51] consider one of the profile and the frontal views and use the Generalized Procrustes Analysis to estimate the two clusters in the shape space. The local texture models (which is similar to ASM) are learned for each cluster separately. During segmentation the updates to the shape are computed using each model and summed with appropriate weights to yield the final segmentation. The parameters of the shape model and shape regularization are performed using EM algorithm.

The strategy we propose here bears some resemblance to the approach of Zhou *et al.* [51]. The majority of aforementioned segmentation approaches require as many facial views as possible in the training set. What we suggest is a way to reduce the training set needed for multi-view face segmentation by applying a recently proposed multiple ASM and multiple AAM fusion algorithm [52]. The idea is to construct a number of models from some predefined facial views, and then segment any view using a model obtained by the weighted fusion of these pre-built models. Note that this is different from the approaches of Lee and Okada [42–44], who decompose the whole range of head motion into a number of subranges, which are in turn approximated by linear subspaces.

In this paper we will consider only horizontal head rotations due to limitations of the landmarked databases we had access to, but the framework can be extended to handle head tilting as well. Following this idea, the models for left, right and frontal views are constructed. Left and right head rotations must not exceed approximately 60° to avoid significant occlusions that induce topology changes in the facial shape (defined by landmarks). Then, given any view of a face, the models are fused with appropriate weights and the face is segmented by the fused model. In

this fashion we limit our training set to only three views.

To evaluate the proposed approach, in the first place, we investigate the ideal case when the optimal fusion weights are known, thus allowing us to measure the potential of the method independently of the weight estimation techniques. Subsequently, a method to estimate the weights is proposed. The experiments demonstrate that the fused model has higher segmentation accuracy than the pre-built models corresponding to the fixed views and the model constructed from all available views. The weight estimation method is tested on the set of manually landmarked images as well as on video sequences.

The remainder of the paper is organized as follows. Section 3.2 briefly describes active shape and appearance models as well as steps required for their fusion. Section 3.3 provides an algorithm for weight estimation for the problem of multi-view face segmentation. Section 3.4 evaluates the proposed method in terms of segmentation and weight estimation accuracy. The paper is concluded by a discussion of some aspects of the approach and conclusions in Sections 3.5 and 3.6, respectively.

3.2 Weighted fusion of several active shape and appearance models

3.2.1 Weighted eigenspace fusion

We will follow the same notation as in [52]. The matrices are written in bold uppercase; vectors are column vectors and written in bold lowercase; the letters with normal typeface denote scalars.

Let us consider M eigenspaces. The i-th eigenspace is computed by Principal Component Analysis (PCA), applied to the set of N_i observations $\mathbb{X}_i = \{\mathbf{x}_{ij} | j = 1, \ldots, N_i\}$. It is defined as a quadruple [20]:

$$\Omega_i = (\bar{\mathbf{x}}_i, \mathbf{\Phi}_i, \mathbf{\Lambda}_i, N_i), \quad i = 1, ..., M \tag{3.1}$$

where the n-vector $\bar{\mathbf{x}}_i$ is the mean of the observations, $\mathbf{\Phi}_i$ is a $n \times m_i$ matrix of eigenvectors, and $\mathbf{\Lambda}_i$ is a $m_i \times m_i$ matrix of eigenvalues (n and m_i have been determined during construction of eigenspaces to be fused). Note that the number of rows of all $\mathbf{\Phi}_i$ is the same (*i.e.,* all the eigenvectors must have the same number of components). Each eigenspace is assigned a weight w_i such that $\sum_{i=1}^{M} w_i = 1$. These weights are used to change the influence of each eigenspace on the fused one. Without loss of generality we shall assume that all the weights are positive. Let

$$p_i = w_i \cdot \left(\sum_{j=1}^{M} w_j N_j \right)^{-1} \tag{3.2}$$

Introducing p_i we transfer the model weights w_i to the observation level, so each observation of i-th model has a weight p_i. Let us define the full observation set as $\mathbb{X} = \bigcup_{i=1}^{M} \mathbb{X}_i$, consisting of $N = \sum_{i=1}^{M} N_i$ observations, with its elements denoted by $\mathbf{z}_k \in \mathbb{X}$ in order to simplify the notation in several formulas (each $\mathbf{z}_k$ is equal to $\mathbf{x}_{ij}$ for some i and j).

The goal of weighted fusion is to compute such an eigenspace $\Omega = (\bar{\mathbf{z}}, \mathbf{\Phi}, \mathbf{\Lambda}, N)$, using the information from Ω_i, $i = 1, \ldots, M$, only, that it is equivalent to the eigenspace computed from the full set of weighted observations $\mathbb{X}$ (should we have had access to them). Here, $\bar{\mathbf{z}}$ is again an n-vector, $\mathbf{\Phi}$ is an $n \times m$ matrix, and $\mathbf{\Lambda}$ is an $m \times m$ matrix, where m is determined during the fusion.

The fusion of the eigenspaces Ω_i, $i = 1, ..., M$ is given by the following steps [52]:

1. Compute the fused mean by

$$\bar{\mathbf{z}} = \sum_{i=1}^{M} N_i p_i \bar{\mathbf{x}}_i \tag{3.3}$$

2. Concatenate column-wise into a matrix $\mathbf{H}$ all the eigenvector matrices $\mathbf{\Phi}_i, i = 1, \ldots, M$, and the differences of all possible pairs of the means $\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j$, where $i, j = 1, \ldots, M$ and $j > i$:

$$\mathbf{H} = [\mathbf{\Phi}_1 | \mathbf{\Phi}_2 | \ldots | \mathbf{\Phi}_M | (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) | (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_3) | \ldots | (\bar{\mathbf{x}}_{M-1} - \bar{\mathbf{x}}_M)] \tag{3.4}$$

3. Orthonormalize $\mathbf{H}$ to get an $n \times p$ basis $\mathbf{Y}$

$$\mathbf{Y} = Orth(\mathbf{H}) \tag{3.5}$$

   where p is determined by an orthonormalization algorithm.

4. Compute the eigenvectors $\mathbf{R}$ and eigenvalues $\mathbf{\Lambda}$ of

$$\frac{1}{1 - \sum_{i=1}^{M} N_i p_i^2} \left[\sum_{i=1}^{M} (N_i - 1) \tilde{\mathbf{D}}_i p_i + \sum_{i=1}^{M} \sum_{j=i+1}^{M} N_i N_j p_i p_j \mathbf{E}_{ij} \right] \tag{3.6}$$

   where

$$\tilde{\mathbf{D}}_i = \left(\mathrm{Y}^T \mathbf{\Phi}_i \right) \mathbf{\Lambda}_i \left(\mathrm{Y}^T \mathbf{\Phi}_i \right)^T, \quad \mathbf{E}_{ij} = \left(\mathrm{Y}^T \left[\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j \right] \right) \left(\mathrm{Y}^T \left[\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j \right] \right)^T \tag{3.7}$$

5. The eigenvalues of the fused eigenspace are given by $\mathbf{\Lambda}$ and the eigenvectors are obtained by $\mathbf{\Phi} = \mathbf{YR}$.

3.2.2 ASM Fusion

An Active Shape Model is constructed from a set of aligned shapes by PCA [7]. Shapes are defined by landmark points placed along the contour of the object of interest. ASM's main component is a Point Distribution Model (PDM) defined for the i-th of M ASMs by:

$$\mathbf{x}_{ij} = \bar{\mathbf{x}}_i + \mathbf{\Phi}_{si}\mathbf{b}_{ij}^s, \quad i = 1, \ldots, M \tag{3.8}$$

where $\mathbf{x}_{ij}$ is a n-vector, representing the j-th shape corresponding to the i-th PDM. It is obtained by concatenating all the landmark coordinates into a single real-valued vector one after another. In other words, if landmarks have coordinates (x_i, y_i) the concatenated vector will be of the form $(x_1, y_1, x_2, y_2, \ldots)^T$. Then, the n-vector $\bar{\mathbf{x}}_i$ is the mean of the aligned shapes in the training set; the $n \times m_i$ matrix $\mathbf{\Phi}_{si}$ and the m_i-vector $\mathbf{b}_{ij}^s$ are the projection matrix and the corresponding projection coordinates, respectively. The values of m_i have been determined during the construction of PDMs to be fused. The letter "s" in subscript or superscript of a symbol relates the latter to the PDM. It must be mentioned that our technique requires that all the PDMs use the same landmark placement and, thus, to have the same number of landmarks.

To fit the model to an image, profiles perpendicular to the contour at landmark positions are used. From pixels sampled along each profile the mean vector and covariance matrix are estimated during the model construction. The collection of such pairs for each landmark constitutes an *Intensity Model*. They are a part of the Mahalanobis distance, which is used to drive the model to the best-fit location during segmentation.

ASM fusion is straightforward. Since each PDM is nothing else than an eigenspace describing the shapes from the training set, the fusion of several ASMs is reduced to the fusion of "aligned" PDM eigenspaces (using the algorithm from Section 3.2.1) and fusion of statistical information for the shape profiles.

To fuse ASMs with weights w_i ($\sum w_i = 1$):

1. Align the means $\bar{\mathbf{x}}_i$ of all PDM's using the Procrustes Analysis estimating the mean according to (3.3).
2. For each $\bar{\mathbf{x}}_i$ a $d \times d$ matrix $\mathbf{S}_i$, that aligns this shape to the mean $\bar{\mathbf{x}}$, is computed. Here d is the dimensionality of landmarks. Note that the shapes are centered at origin so the matrix accounts only for rotation and scaling.
3. Construct a matrix $\mathbf{\Xi}_i$, which is a $\frac{n}{d} \times \frac{n}{d}$ block-diagonal matrix with repeating $\mathbf{S}_i$ along its diagonal:

$$\mathbf{\Xi}_i = \begin{bmatrix} \mathbf{S}_i & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{S}_i \end{bmatrix} \tag{3.9}$$

4. Fuse PDMs by applying the eigenspace fusion scheme to eigenspaces corresponding to the aligned means $\Xi_i \bar{\mathbf{x}}_i$ and eigenvectors $\Xi_i \mathbf{\Phi}_{si}$.

5. Fuse the Intensity Models. For each landmark:

   (a) Fuse the mean profile using (3.3)

   (b) The fused covariance matrix is given by

$$\frac{1}{1 - \sum_{i=1}^{M} N_i p_i^2} \left[\sum_{i=1}^{M} (N_i - 1)\, \mathbf{D}_i p_i + \sum_{i=1}^{M-1} \sum_{j=i+1}^{M} N_i N_j p_i p_j \left(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\right) \left(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\right)^T \right] \tag{3.10}$$

3.2.3 AAM Fusion

Let us assume that we are given M AAMs and that each model has an associated weight w_i such that $\sum w_i = 1$. The first component of an AAM that we are going to consider is a PDM, which describes the shape variation of the object of interest. The classical linear PDM used in AAM is again defined by (3.8). The eigenspace associated with each PDM is $\Omega_{si} = (\bar{\mathbf{x}}_i, \mathbf{\Phi}_{si}, \mathbf{\Lambda}_{si}, N_i)$.

The next component of an AAM is a Texture Model (TM). TMs are constructed from the intensity values of pixels inside the shape.

A linear TM is defined by

$$\mathbf{g}_{ij} = \bar{\mathbf{g}}_i + \mathbf{\Phi}_{gi} \mathbf{b}_{ij}^g, \quad i = 1, \ldots, M \tag{3.11}$$

where the k_i-vector $\mathbf{g}_{ij}$ is the j-th texture instance for the i-th TM, the k_i-vector $\bar{\mathbf{g}}_i$ is the mean texture for i-th model, the $k_i \times l_i$ matrix $\mathbf{\Phi}_{gi}$ is the matrix of eigenvectors, and $\mathbf{b}_{ij}^g$ are the l_i projections of $\mathbf{g}_{ij}$ in the subspace spanned by $\mathbf{\Phi}_{gi}$. The corresponding eigenspaces are $\Omega_{gi} = \left(\bar{\mathbf{g}}_i, \mathbf{\Phi}_{gi}, \mathbf{\Lambda}_{gi}, N_i\right)$. The values k_i and l_i have been determined during the construction of the TMs to be fused. The letter "g" in subscript or superscript of a symbol relates the symbol to the TM.

Having parameterized shape and texture, a combined AAM is constructed and defined by:

$$\mathbf{b}_{ij} = \mathbf{\Phi}_{ci}\, \mathbf{c}_{ij}, \quad i = 1, \ldots, M; \quad j = 1, \ldots, N_i \tag{3.12}$$

with a $(m_i + l_i) \times q_i$ eigenvector matrix $\mathbf{\Phi}_{ci}$. The $(m_i + l_i)$-vector $\mathbf{b}_{ij}$ is constructed from the corresponding i-th TM parameters and i-th PDM parameters as follows:

$$\mathbf{b}_{ij} = \begin{bmatrix} \mathbf{W}_{ci} \cdot \mathbf{b}_{ij}^s \\ \mathbf{b}_{ij}^g \end{bmatrix} = \widetilde{\mathbf{W}}_{ci} \cdot \begin{bmatrix} \mathbf{b}_{ij}^s \\ \mathbf{b}_{ij}^g \end{bmatrix}; \quad \widetilde{\mathbf{W}}_{ci} = \begin{bmatrix} \mathbf{W}_{ci} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$$

$\mathbf{W}_{ci}$ being a diagonal $m_i \times m_i$ matrix of weights, calculated from the eigenvalues of the PDM and TM, used to make shape and texture parameters commeasurable [8]. The corresponding eigenspaces are defined by $\Omega_{ci} = (\mathbf{0}, \mathbf{\Phi}_{ci}, \mathbf{\Lambda}_{ci}, N_i)$. The value q_i has been determined during construction of the i-th AAM's combined model. The letter "c" in subscript or superscript of a symbol relates the symbol to the combined model.

Matching the above combined AAM to an image is performed in an iterative manner using the prediction matrices calculated during model construction. The following equations are used to update model parameters:

$$\mathbf{R}_c \cdot \mathbf{r}\left(\mathbf{c}^*, \mathbf{t}^*\right) = \Delta\mathbf{c}, \quad \mathbf{R}_t \cdot \mathbf{r}\left(\mathbf{c}^*, \mathbf{t}^*\right) = \Delta\mathbf{t} \tag{3.13}$$

where $\Delta\mathbf{c} = \mathbf{c} - \mathbf{c}^*$ and $\Delta\mathbf{t} = \mathbf{t} - \mathbf{t}^*$ are the differences in model and pose parameters, $\mathbf{c}^*$ and $\mathbf{t}^*$ are current estimates of the parameters and $\mathbf{r}\left(\mathbf{c}^*, \mathbf{t}^*\right) = \mathbf{g}_{image}\left(\mathbf{c}^*, \mathbf{t}^*\right) - \mathbf{g}_{model}\left(\mathbf{c}^*, \mathbf{t}^*\right)$ is the texture residual (*i.e.,* the difference between the texture sampled from the image under the shape generated by the model with parameters $\mathbf{c}^*$, $\mathbf{t}^*$ and the texture generated by the model with the same parameters). The matrices $\mathbf{R}_c$ and $\mathbf{R}_t$ are the inverse of the Jacobians that are calculated during model building.

Fusing Active Appearance Models requires the following steps [52]:

1. *PDM Fusion.* Align the mean shapes $\bar{\mathbf{x}}_i$ by Procrustes Analysis. Compute the fused mean shape $\bar{\mathbf{x}}$ and the transformations Ξ_i (3.9) that align $\bar{\mathbf{x}}_i$ to $\bar{\mathbf{x}}$ for every i-th model. Fuse $\Omega_{si} = \left(\Xi_i\bar{\mathbf{x}}_i, \Xi_i\mathbf{\Phi}_{si}, \mathbf{\Lambda}_{si}, N_i\right), i = 1, \ldots, M$ to obtain a fused PDM $\Omega_s = (\bar{\mathbf{x}}, \mathbf{\Phi}_s, \mathbf{\Lambda}_s, N)$.

2. *Texture Model Fusion.*

   (a) Calculate the spatial warps $\tau_i\left(\bar{\mathbf{g}}_i\right)$ and $\tau_i\left(\mathbf{\Phi}_{gi}\right)$, such that the i-th warp corresponds to a mapping of the k_i-dimensional texture vector from the mean shape $\bar{\mathbf{x}}_i$ of the i-th PDM (3.8) to the k-dimensional texture vector, corresponding to the mean shape $\bar{\mathbf{x}}$ of the fused PDM. Note that $\tau_i\left(\mathbf{\Phi}_{gi}\right)$ is a matrix whose columns are the columns of $\mathbf{\Phi}_{gi}$ warped by τ_i.

   (b) Fuse $\widetilde{\Omega}_{gi} = \left(\tau_i\left(\bar{\mathbf{g}}_i\right), \tau_i\left(\mathbf{\Phi}_{gi}\right), \mathbf{\Lambda}_{gi}, N_i\right)$, $i = 1, \ldots, M$ using the proposed eigenspace fusion algorithm to obtain the fused TM $\Omega_g = \left(\bar{\mathbf{g}}, \mathbf{\Phi}_g, \mathbf{\Lambda}_g, N\right)$.

3. *Fusion of the Combined Models.*

   (a) Compute $\mathbf{W}_c = \mathbf{I} \cdot \sqrt{\frac{tr\left(\mathbf{\Lambda}_g\right)}{tr\left(\mathbf{\Lambda}_s\right)}}$.

   (b) Fuse the eigenspaces $\widetilde{\Omega}_{ci} = (\mathbf{0}, \mathbf{\Psi}\mathbf{\Gamma}_i\mathbf{\Phi}_{ci}, \mathbf{\Lambda}_{ci}, N_i)$, where

$$\mathbf{\Psi} = \begin{bmatrix} \mathbf{W}_c\mathbf{\Phi}_s^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \mathbf{\Gamma}_i = \begin{bmatrix} \mathbf{\Phi}_{si}\mathbf{W}_{ci}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Phi}_g^{-1} \cdot \tau_i\left(\mathbf{\Phi}_{gi}\right) \end{bmatrix} \tag{3.14}$$

4. Fuse the prediction matrices $\mathbf{R}_{ci}$ and $\mathbf{R}_{ti}$, $i = 1, \ldots, M$ of all the AAMs according to the following formulas:

$$\begin{aligned} \mathbf{R}_t &= \left[\sum_{i=1}^{M} N_i p_i \cdot \tau_i \left(\mathbf{R}_{ti}^{-1} \right) \right]^{-1} \\ \mathbf{R}_c &= \left[\left(\sum_{i=1}^{M} N_i p_i \cdot \tau_i \left(\mathbf{R}_{ci}^{-1} \right) \left(\mathbf{\Psi \Gamma}_i \mathbf{\Phi}_{ci} \right)^{-1} \right) \cdot \mathbf{\Phi}_c \right]^{-1} \end{aligned} \tag{3.15}$$

3.3 Weight estimation for multi-view face segmentation

In the previous section we have briefly described a framework for the fusion of Active Shape and Appearance Models. From now on we will concentrate on its application to multiview facial analysis. The goal of this work is to show how fusion can be used to reduce the training set of the models and how to extend the capabilities of classical 2D ASMs and AMMs to handle views absent in the training set. In a typical scenario these models have to be trained with many views of a face in order to segment any facial view. In this study we would like to investigate whether it would be possible to limit the required views to frontal and lateral only while still be able to perform the segmentation of any intermediate view. The proposed method relies on the fusion of models corresponding to these three views with an optimal weight.

In this section we would like to describe one of the possible approaches for weight estimation based on iterative segmentation. The idea is to find such a weight that, after segmenting an image with the fused model, the difference between the modeled and the sampled textures is minimized (in the least squares sense). A similar approach has been adopted by Cootes *et al.* [47] to determine which AAM is best suited for each particular image. We will treat only the case of AAM, for ASMs, as it will be demonstrated, are not very sensitive to accurate weight estimation.

Let us construct three AAMs from three sets of views: frontal view, left and right views with 60° head rotation each. They will be referenced by `frontal`, `l60` and `r60` views and models, respectively.

Let $w \in [-1, 1]$. Let M_F, M_L and M_R denote the `frontal`, `l60` and `r60` models. Since during head rotation the head rotates either to the left or to the right there is no need to fuse all three models simultaneously, therefore let us formulate the fused model as follows:

$$M(w) = \begin{cases} [(-w) \otimes M_R] \oplus [(1 - |w|) \otimes M_F] & , w < 0 \\ [(1 - |w|) \otimes M_F] \oplus [w \otimes M_L] & , w \geq 0 \end{cases} \tag{3.16}$$

where w is the weight, "$\otimes$" represents weighting the model and "$\oplus$" represents fusion. As we can see the problem of finding the optimal model for segmenting

a specific facial pose is reduced to optimization of a function of one parameter, varying from -1 to 1. When the fused model is used to segment a specific image, the result is the shape of the face (the contours, defined by landmarks) and two texture vectors: one is the real texture inside the shape, sampled from a given image and normalized, and the other is the texture estimated by the model as the best matching facial appearance. The objective of optimization is to find a weight that minimizes root mean square error (RMSE) between these two texture vectors. It is not an easy task to formulate the gradient for such a function, so it was decided to use an optimization algorithm without derivatives [53, p. 72], that combines golden search and successive parabolic interpolation to avoid some local minima.

The weight estimation can be summarized by the following steps:

Algorithm 1: Pose-independent model-based segmentation with simultaneous fusion weight estimation.

Data: Models M_L, M_R, M_F, image with a face
Result: Fitted shape and texture, optimal weight
Initialize the weight w_0 according to the optimization algorithm of [53];
k = 0;
while *not converged* **do**
  Fuse M_L, M_R, M_F with the weight w_k according to (3.16);
  Match the fused model $M(w_k)$ to the image;
  Compute the error between the sampled and modeled textures;
  $w_{k+1} = w_k$ updated according to the optimization algorithm;
  k = k + 1;
end

3.4 Experiments

For our tests we have used the AV@CAR [54] database (image size 768×576). From this database we have chosen manually landmarked images of seven different facial views corresponding to 0°, $\pm 20^\circ$, $\pm 40^\circ$, $\pm 60^\circ$ horizontal head rotations of forty subjects. Larger angles were not considered as many landmarks become occluded and the model required for its analysis would have a different topology. Views corresponding to left rotations can be seen in Fig. 3.1.

Before proceeding, to simplify the description of experiments, we would like to introduce a number of conventions. Firstly, when speaking about a model we will mean, unless specified otherwise, both AAM and ASM models as most of the considerations equally apply to both of them. Secondly, the considered dataset is separated into seven subsets according to the angle. We shall call them `frontal`, `l20`, `l40`, `l60`, `r20`, `r40`, `r60` where "`l`" and "`r`" mean left and right rotations and

Figure 3.1: Sample images of the frontal and three left views used for model construction and testing.

the number stands for the corresponding rotation angle. We will use the same names for the models constructed from the corresponding datasets (*e.g.*, `frontal` model, `l60` model). And thirdly, we are always going to fuse the `frontal` model with either `l60` or `r60`, so M_R in (3.16) corresponds to `r60` and M_L to `l60`. The model matching always starts from the mean model instance rescaled to fit into the smallest rectangle containing the face. The rectangle is defined manually but could be estimated as output of any face detection algorithm [55, 56]. Whenever the manually landmarked shapes are available, the accuracy of segmentation is evaluated using the point-to-point error:

$$\varepsilon = \frac{d}{n} \sum_{i=0}^{n/d-1} \sqrt{\sum_{j=1}^{d} \left(\hat{x}_{i \cdot d+j} - x_{i \cdot d+j} \right)^2} \tag{3.17}$$

where x_i is the i-th element of the fitted shape, n-vector $\mathbf{x}$, and $\hat{x}_i$ is the i-th element of the manually defined shape $\hat{x}$, and d is the number of dimensions as defined in Section 3.2.2.

In the following sections we will evaluate the fusion framework in itself, when the optimal weight is known (to separate the error introduced by the weight estimation), and altogether with the weight estimation scheme.

3.4.1 Fusion framework evaluation with a known optimal weight

In the first experiment we want to demonstrate how the fusion of models can be used to segment the views absent in the training set. To that end, the optimal fusion weights for `l20`, `r20`, `l40` and `r40` datasets have been determined. To determine these weights all the images have been segmented by the model $M(w)$ with w varying from -1 to 1 in steps of 0.05. For each image, the weight resulting in the smallest point-to-point error with respect to the manual landmarks was declared

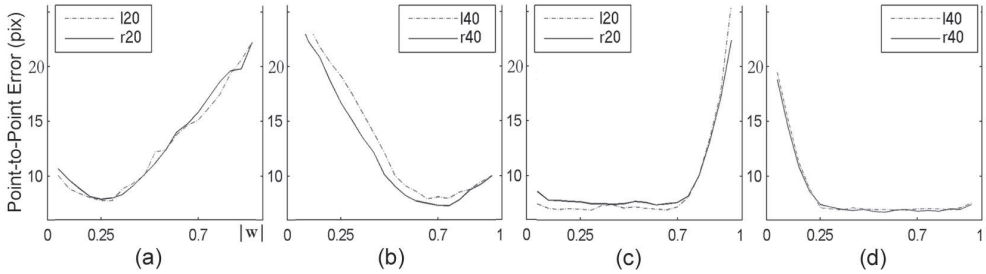


Figure 3.2: The influence of weight estimation on point-to-point segmentation error. Plotted are the average point-to-point error curves of segmenting the `l20`, `r20`, `l40` and `r40` sets by the fused AAM (a)-(b) and ASM (c)-(d) for different weights against $|w|$. Each plot shows pairs of errors corresponding to left and right view. The errors are estimated with respect to manual landmarks.

optimal. Therefore, a sequence of weights was obtained for each dataset. The optimal weight for each dataset was estimated by averaging the weights from each sequence. The graphs of all the error-weight relationships for all four views and both ASM and AAM models are presented in Fig. 3.2. The error is plotted against the absolute value of the weight to verify the similarity of the graphs corresponding to left and right views. It can be noted that ASM appears to be much less sensitive to the accuracy of weight estimation. The AAM in its turn demonstrates no significant reduction in segmentation accuracy within ± 0.05 interval around the optimal weight. As it can be seen from that figure (in the case of AAM) the optimal weights are ± 0.7 for `l40` and `r40` sets and ± 0.25 for `l20` and `r20` sets.

Fig. 3.3 presents a comparison of accuracy of segmentation performed according to various strategies with 95% confidence intervals. The meaning of the labels depends on the testing set and is presented in Table 3.1. Each cell of this table explains how the model is constructed for each particular testing set.

The point-to-point errors are computed with respect to manual landmarks. It is worth to note that `normal` is the typical approach when all the available views are used to train the model (in this case only right and frontal or left and frontal; using all three of them significantly distorts the mean shape and the segmentation error is quite large), and `closest` is the same approach as that of Cootes *et al.* [47]. As a reference, the figure shows the `baseline` segmentation results obtained by models constructed from the test sets themselves, thus providing the best possible results. It can be seen that, in all the cases, the model obtained by the fusion performed better or in the case of ASM equally well as the best of the other models (except the `baseline`). An example of segmentation using each of the mentioned models can be seen in Fig. 3.4.

Considering the presented results we can conclude that:

Table 3.1: The types of the evaluated models

Model Type	Test Set			
	`l40`	`l20`	`r20`	`r40`
Normal	Single model trained on `frontal` and `l60` sets	Single model trained on `frontal` and `l60` sets	Single model trained on `frontal` and `r60` sets	Single model trained on `frontal` and `r60` sets
Fused	Fusion of `frontal`, `l60` and `l60` models with the weight equal -0.7	Fusion of `frontal`, `l60` and `l60` models with weight equal -0.25	Fusion of `frontal`, `l60` and `l60` models with the weight equal 0.25	Fusion of `frontal`, `l60` and `l60` models with the weight equal 0.7
Closest	`l60` model	`frontal` model	`frontal` model	`r60` model
Baseline	`l40` model	`l20` model	`r20` model	`r60` model

- Using different weights the fusion provides a way to linearly "interpolate" active shape and appearance models. The approach when the `closest` model is chosen could be considered as zero-order or nearest neighbor interpolation.
- The segmentation of any facial view, corresponding to a horizontal head rotation, can be improved by fusion of the two closest views, provided that the weights are estimated correctly. In this particular case, having only three models corresponding to frontal, left and right views, fusion could be used to interpolate these models and use the result to segment any other view.
- Active Appearance Models are much more sensitive to the correct weight estimation. Which means that while for ASM it would be enough to fuse `frontal` and `l60` models with equal weights to segment any left view, AAM requires more accurate weight estimation.

3.4.2 Fusion framework evaluation with unknown optimal weight

In this section we would like to evaluate how the fusion benefits the accurate segmentation when the weight has to be estimated, using the approach proposed in Section 3.3. The goal is to find a weight for the model $M(w)$ in (3.16) such that it provides the best possible reconstruction of the facial texture after segmentation. In other words, the RMSE between the sampled and generated textures should be minimal. An example of this objective function for all the sets has been plotted

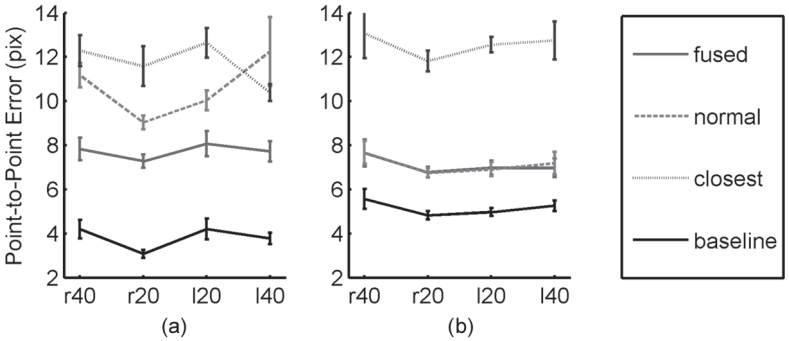


Figure 3.3: Comparison of accuracy of different segmentation approaches for known optimal fusion weight. Plotted are the average point-to-point errors of segmenting the testing sets by the fused AAM (a) and ASM (b) with 95% confidence intervals. The errors are estimated with respect to manual landmarks.

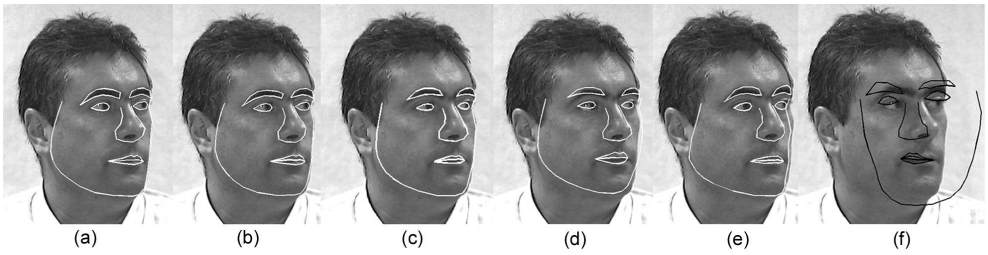


Figure 3.4: An example of segmenting one of the `140` images using different AAM models. (a) manual segmentation; (b) best possible segmentation with AAM built from the testing set; (c) fused AAM with the optimal weight; (d) AAM built from the `frontal` and `160` sets; (e) `160` AAM; (f) `frontal` AAM.

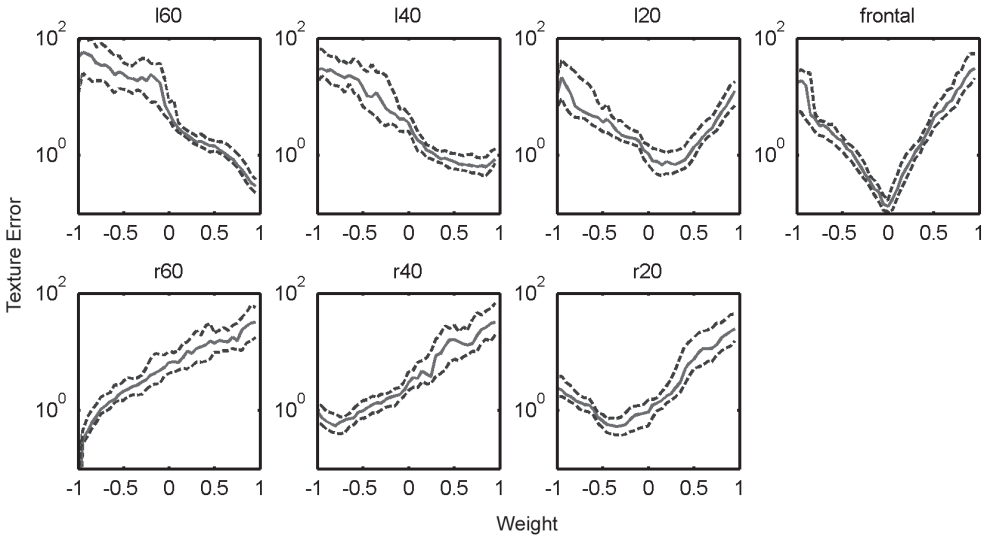


Figure 3.5: Texture errors of segmentation by a fused AAM (3.16), computed for the weight varying from -1 to 1 in steps of 0.05. Plotted are the 1st, 2nd (median) and 3rd quartiles for all the sets. The ordinate is scaled logarithmically.

Table 3.2: Percentages of correctly estimated weights

Set	`l20`	`r20`	`l40`	`r40`	`l60`	`r60`	`frontal`	Total
0.05	24	34	22	32	30	30	35	207 (73.9%)
0.10	32	34	27	33	36	30	38	230 (82.2%)
Diverged	4	0	4	3	5	9	0	25 (8.9%)

in Fig. 3.5 on a logarithmic scale. The error is computed for each weight sampled from the interval $[-1, 1]$ in steps of 0.05. It can be seen that these functions are not strictly unimodal, nevertheless the minimization algorithm is robust to certain local minima [53].

Firstly, to evaluate the accuracy of the weight estimation, the algorithm was applied to all the landmarked sets: `frontal`, `l20`, `l40`, `l60`, `r20`, `r40`, `r60`. For every image the estimated weight was compared to the optimal value, estimated by exhaustive search. The distributions of the absolute differences between the optimal and estimated weights are presented in Fig. 3.6. Table 3.2 shows how many images from the data sets (40 images in each) had the weight estimation error smaller than 0.05 and 0.10. The last row shows the number of images where the model diverged completely due to incorrect weight estimation. The boxplot of the point-to-point errors corresponding to the Table 3.2 is shown in Fig. 3.7a, where crosses (outliers)

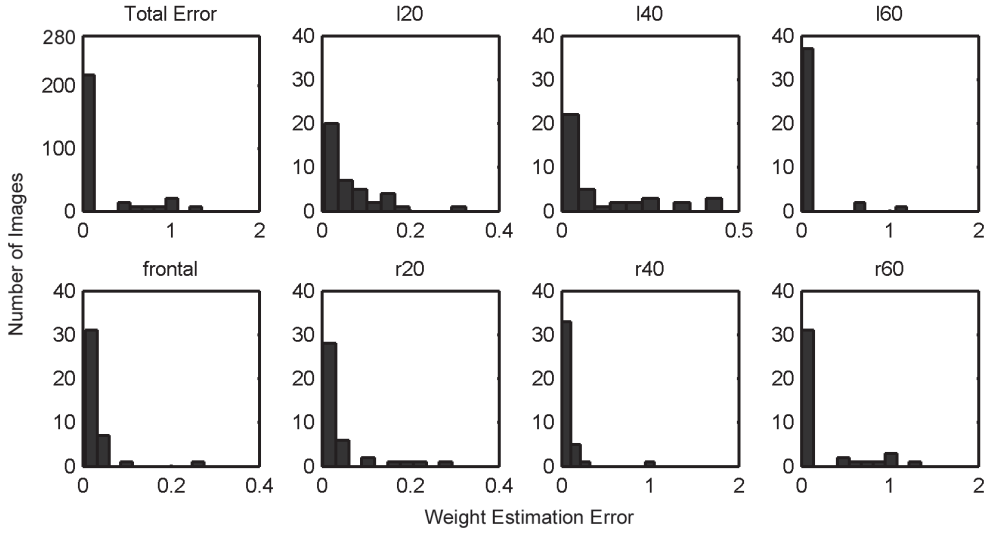


Figure 3.6: Histograms of weight estimation errors per testing set for AAM fusion. The errors are computed with respect to the optimal weights, estimated by exhaustive search.

correspond to the cases of divergence. In general it can be seen that the most successful estimations were achieved for the frontal images. On the other hand in some cases when the weight estimation error was slightly larger than 0.1 the model still was able to correctly segment the images.

Finally we evaluate the algorithm on several video sequences. Since in these, the faces have not been manually delineated, the only quantitative way to evaluate the performance is to show the accuracy of weight estimation. To make the plots more meaningful we investigated the relationship between the weight and the angle. In other words an optimal weight has been determined, by an exhaustive search, for every image in every set: `frontal`, `l20`, `l40`, `l60`, `r20`, `r40`, `r60`. The average optimal weight with its 95% confidence interval for every available angle is shown in Fig. 3.7b. As it can be seen the relationship is approximately linear.

To evaluate the algorithm on video sequences, seven sequences (about 17 frames each) of horizontal head rotation have been taken from the AV@CAR database. Each video frame has been segmented separately. For each frame the optimal weight for fusing the three AAMs was estimated by exhaustive search as described in Section 3.3. But in this case, in the absence of manual landmarks, the weight corresponding to the smallest texture error was considered optimal. The fused model was initialized as in Section 3.4.1 by fitting the mean model instance into the manually defined rectangle which contained the face. On average the convergence of the

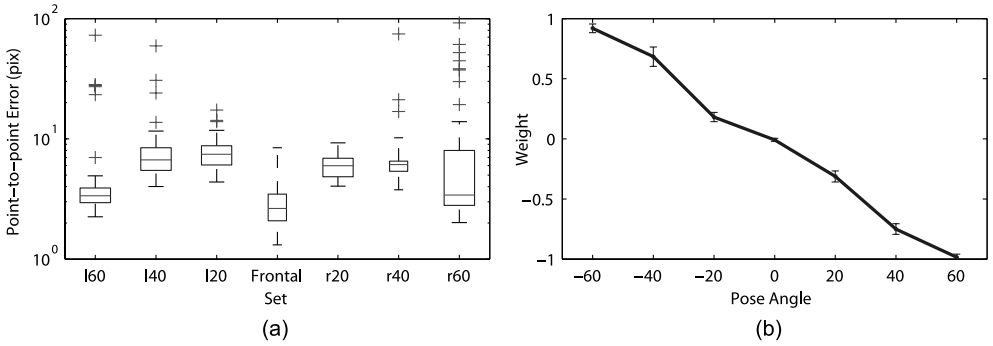


Figure 3.7: (a) Evaluation of segmentation accuracy by the fused AAM in terms of the point-to-point segmentation errors on the testing sets with respect to manual landmarks, the size of each image is 768×576. (b) Relation between the fusion weight and pose angles, with error bars representing 95% confidence intervals of the mean.

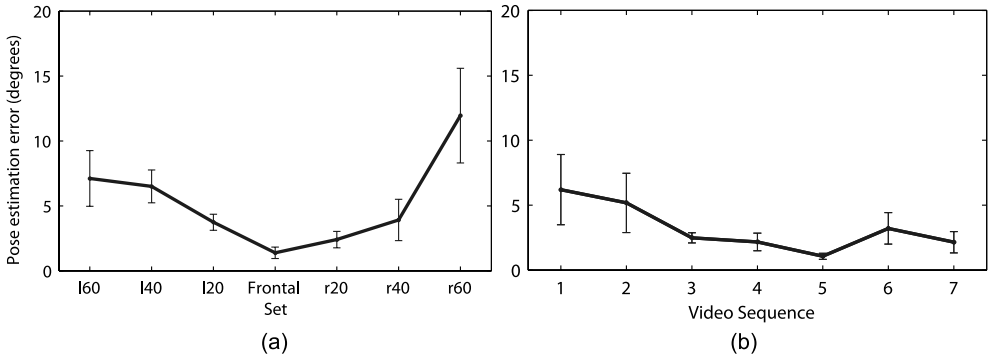


Figure 3.8: Evaluation of the pose estimation accuracy (assuming linear relationship between the weight and pose angle) on testing sets (a) and video sequences (b). The pose errors are computed as the difference between the optimal and estimated weight multiplied by 60. The optimal weight is estimated by exhaustive search. The size of each image and video frame is 768×576. Error bars represent 95% confidence interval of the mean.

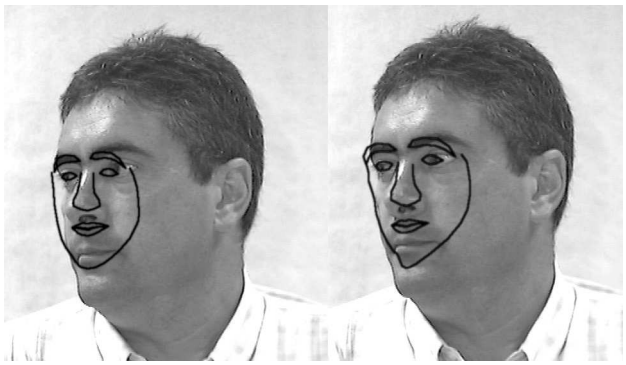


Figure 3.9: Sample video frames wherein the AAM has diverged.

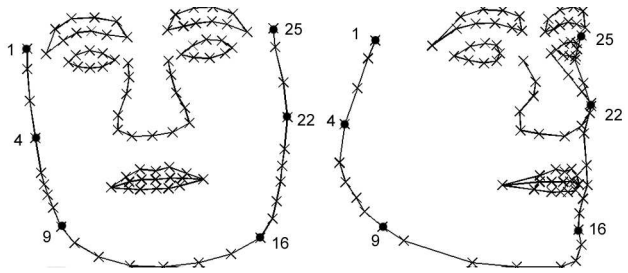


Figure 3.10: Landmark correspondence between the frontal and profile views. The landmarks that do not correspond to prominent facial features stay approximately in the same relative position with respect to the prominent ones.

optimization scheme was achieved in 8 iterations.

In order to evaluate the accuracy of weight estimation on video sequences, the optimal weight for each frame was compared to that determined by the optimization algorithm. The resulting weight estimation errors are plotted in Fig. 3.8b. For a reference, Fig. 3.8a shows the weight estimation accuracy on the testing set of the static images. Note that these plots use the linearity of the relationship between the angle and the weight to map the weight difference to angle difference. The model diverged only in 2 frames of one video sequence corresponding to an extreme rotation to the right, the corresponding weight estimation errors were 0.6 and 0.7 (the minimization converged to a local minimum near $w = 0$). These two frames can be seen in Fig. 3.9.

Figure 3.11: Texture obtained by AAM segmentation: (a) an original from the `120` set and the matched model; (b) an original from the `140` set and the matched model.

3.5 Discussion

We would like to address a number of issues directly related to the problem of multi-view face segmentation from images.

The first one is landmark placement. The principal role of landmarks is to define contours of a face and its features (eyes, nose, mouth, *etc.*). It can be noted that, as the head rotates, some landmarks, if maintained fixed, would become occluded. At the same time the visible contours of the face changes. Therefore, during head rotations the landmarks must stay on the contour of the features they outline (just as it is done in many other studies [47, 48]). Since both ASM and AAM model the displacements of points, we tried to keep the number of irrelevant displacements (along the contour) to the minimum by keeping the landmarks that do not correspond to the prominent facial features approximately in the same relative position with respect to the prominent ones. Fig. 3.10 illustrates the correspondence between landmarks in two facial views.

Another problem is the absence of texture information in occluded areas. Since there is only one camera, there is no way of getting information about the occluded cheek of the rotated head. Therefore, when the texture is warped from the rotated face to the mean shape of the fused model, there will be texture distortions. The views with more occlusions will have their texture stretched depending on the rotation angle between that view and the view corresponding to the mean shape. In spite of this problem, AAMs recover the texture quite well and most facial features are still distinguishable as demonstrated in Fig. 3.11. Of course this effect can be reduced by including more views of faces but that is contrary to our goal, which was to reduce the number of training views.

Now we would like to put our proposed method in the context of related work.

Among the existing approaches to pose-independent 2D face segmentation based on AAM and ASM there are several approaches that are worth noting. Cootes *et al.* [47] constructed separate models for a number of viewpoints and segmented a given face by the model corresponding to the closest viewpoint. This case was considered in Section 3.4.1, where it was shown that this approach can be improved by fusion without requiring additional training data. The authors also proposed another approach, the Coupled-View Appearance Models, where the model is constructed from pairs of different facial views taken simultaneously. But the latter approach requires that all the facial views are captured simultaneously. The papers by Gross *et al.* [48] and Hu *et al.* [57] proposed alternative approaches, one based on a modified AAM matching algorithm and the other using an AAM based on wavelets. All of these approaches require more than just frontal and two lateral views for training. Our goal was to develop a strategy for model interpolation that allows creating facial appearances unavailable during training. As a consequence the training set can be reduced to only a minimum set of views. Another paper, by Zhou *et al.* [51] reported all the errors as difference between the methods giving the percentages of images where their method outperformed the other ones.

In spite of the aforementioned difficulties, we can compare our method to that of Wan *et al.* [49] who used the ORL face database [58] for evaluation. Since the results are reported in pixel unit, we expressed the results as percentage of average inter-eye distance (distance between eye centers). In AV@CAR database this distance is approximately 111 pixels and in the ORL – 35 pixels. It should be noted as well that most subjects from the ORL database have only two lateral views (left and right) per face, each corresponding to a rotation angle smaller than 45°. The normalized segmentation accuracy results, presented in Table 3.3, seem to be comparable for both algorithms, although those corresponding to the frontal view are substantially better in our case. On the other hand it should be noted that our model was trained on 60°-rotated and frontal faces, and tested on other facial views of the same people, while Wan *et al.* [49] used the same views for both training and testing, but corresponding to different people. Thence, in the latter case it can be concluded that the view-specific information has been learned by the model and it is not clear how it would handle views unavailable during the training.

Our evaluation is limited to only horizontal head rotations. That is related to the unavailability of landmarked multiple-view databases. Therefore the approach was adapted to that particular problem, nevertheless it can be easily generalized to any head pose by minimizing the following function

$$M(w_1, w_2) = M_{UD}(w_1) \oplus M_{LR}(w_2) \oplus (1 - |w_1| - |w_2|) \otimes M_F \tag{3.18}$$

Table 3.3: Segmentation accuracy comparison between different algorithms in terms of point-to-point error

Our approach		**Wan *et al.* [49]**	
Set	**Mean Error(%)**	**Set**	**Mean Error(%)**
`l20`	7.1	`left`	7.6
`l40`	8.3		
`r20`	5.5	`right`	6.3
`r40`	7.5		
`frontal`	2.6	`frontal`	7.1

where

$$M_{UD}(w) = \begin{cases} -w \otimes M_U & , w < 0 \\ w \otimes M_D & , w \geq 0 \end{cases} \tag{3.19}$$

$$M_{LR}(w) = \begin{cases} -w \otimes M_R & , w < 0 \\ w \otimes M_L & , w \geq 0 \end{cases} \tag{3.20}$$

and M_U, M_D are the models constructed from the faces looking upwards and downwards, respectively. To optimize this function, any minimization algorithm that does not require analytic derivatives or any derivative at all can be used. One possible candidate is Powell's algorithm (an alternative could be evolutionary algorithms). Most databases have only faces looking strictly upwards and downwards without intermediate angles. Evaluating our framework on extreme views would be equivalent to evaluating the standard AAM (since we fuse models corresponding to these extreme views).

Finally, we would like to comment on performance. As it was mentioned, it takes about eight iterations to converge to the optimal weight and to segment an image. Normally, it takes a couple of minutes to perform these eight matchings and fusions (for 768×576 images with the face occupying approximately a rectangular area of 240×200 pixels) with our non-optimized matching routines. But if the images are reduced four times to a still reasonable size, when the face occupies approximately 60×50 pixels the whole process takes about 5 seconds (on the Intel Pentium Q6600 2.40GHz). It is worth to mention that the proposed optimization approach relies on interval partitioning and since all the possible partitionings are easily predictable, all the models can be fused *a priori* and stored. In this case the whole segmentation takes about 0.4 seconds (including loading and saving data).

3.6 Conclusions

In this work we have presented an application of AAM and ASM fusion to multi-view face segmentation. The fusion can be casted into a model interpolation problem, allowing to obtain a better segmentation for views absent in the training set. The latter leads to a possibility of reducing the amount of manually landmarked facial views required for training and keeping them to a minimum: frontal and two lateral (60°) facial views. Then if the fusion weight is estimated correctly, any facial view can be segmented by the fused model. In Section 3.3 we presented a simple algorithm for weight estimation. The estimation failed only in 8.9% of 280 testing images, which converged to incorrect local minima resulting in incorrect segmentation. Since each image was segmented independently the weight estimation in video sequences could be significantly improved by tracking the weight dynamics along the sequence. As the future work we would like to test our methodology on the CAS PEAL R1 [59] database.

CHAPTER 4

Left-ventricular Epi- and Endocardium Extraction from 3D Ultrasound Images Using an Automatically Constructed 3D-ASM

Abstract - *There is great interest in automating the diagnosis of cardiac pathologies through segmentation. Majority of the proposed algorithms in 3D ultrasound (3DUS) cover only left-ventricular (LV) endocardium analysis. Here, we propose an automatic method for constructing an Active Shape Model (ASM) to segment the complete LV. The automatic construction of the Point Distribution Model, a part of the ASM, has been already addressed in the literature and can be handled through image registration. But high level of noise and poor spatial resolution hampers the direct application of these techniques to 3DUS. Therefore, to automatically construct an ASM for US segmentation, we constructed the PDM from multidetector computed tomography data where the registration is much more accurate and robust. To automatically learn the appearance of the US images we have used artificially generated ones using two approaches: one that assumes a uniform point spread function and does not take into account the geometry of the transducer, and a more comprehensive one, implemented in Field II Matlab toolbox. The epi- and endocardium segmentation accuracy of our ASM was evaluated on 20 cardiac resynchronization therapy patients. Apart from accuracy evaluation, we also show that for ASM training it is beneficial to use the simple US modeling technique which is fast and avoids costly manual landmarking.*

Adapted from C. Butakoff, S. Balocco, F.M. Sukno, C. Hoogendoorn, C. Tobón-Gómez, G. Avegliano, A.F. Frangi. Left-ventricular Epi- and Endocardium Extraction from 3D Ultrasound Images Using an Automatically Constructed 3D ASM. *Medical Image Analysis*, submitted, 2009.

4.1 Introduction

Ultrasound (US) is known to be the fastest, least expensive and least invasive screening modality for imaging the heart. Because of the 3D structure and deformation of the heart muscle during the cardiac cycle, analysis of irregularly shaped cardiac chambers or description of valve morphology using 2D images is inherently limited. Developments in 3D echocardiography started in the late 1980s [60]. During the last two decades it evolved from free-hand scanning, replaced later by mechanical scanning of several planes using a linear transducer, to matrix phased-array transducers that are able to acquire a 3D volume of the whole heart almost in real time.

The appearance of this new modality brought in new challenges and the need for new analysis tools, many of which rely on correct segmentation of the myocardium. However, the quality of the data is not sufficient yet, essentially due to poor spatial resolution of the hardware. As a matter of fact, the suboptimal quality forced many studies to reject up to one third of the data [61, 62]. An extensive survey of traditional approaches to ultrasound segmentation can be found in Noble *et al.* [63], Frangi *et al.* [64], Lelieveldt *et al.* [65], and Angelini *et al.* [66, 67]. The classifications of approaches to modeling cardiac geometry can be found in Montagnat *et al.* [68] and Frangi *et al.* [64].

Let us start with approaches using explicit surface representation. To introduce a shape constraint on a segmentation algorithm, Hong *et al.* [69] proposed to use a set of prototype shapes. The resulting shape is a Nadaraya-Watson kernel-weighted average of the prototypes. The authors propose to use 2D Haar-like features to detect the myocardium and require manual annotation in four-chamber view to reduce the search space of the optimization algorithm. The alignment of the prototypes uses as well the known apical four-chamber view (A4C) plane. To compute the 2D Haar-like features, the authors propose to cut the 3D volume into several long-axis slices. It might be useful to note that the segmentation accuracy is evaluated in voxels, and their conversion to millimeters is not straightforward.

A fully automatic registration-assisted segmentation approach with a wiremesh was proposed in Zagrodsky *et al.* [70]. Rigid registration based on mutual information is used to initialize the segmentation. External forces are generated using a 3D extension of the Sobel edge detector with clamped intensities to remove strong and weak edges. Subsequently the zone of influence of the edges is enlarged by a 3D extension of the generalized gradient vector field.

Ping *et al.* [71] proposed an active contour approach on a multilevel cubic B-spline grid for segmenting both epi- and endocardium on echocardiographic images with contrast agent. The movement of the contour is performed by displacing vertices of the grid. The segmentation combines fuzzy feature information and a multilevel freeform deformation model into the objective function that has to be minimized in order to obtain accurate segmentation. This is the only study on Real Time 3D

(RT3D) ultrasound data that demonstrates segmentation of epicardium. The idea is to segment the endocardium and use it as a constraint for epicardium segmentation. The essential drawback of the paper is that the algorithm is evaluated only on four ultrasound volumes with contrast agent and the results are provided in terms of the overlap between the true shapes and those obtained by segmentation, giving thus no possibility to compare it to other studies.

A segmentation of triplane echocardiograms by 2D constrained active appearance motion models (AAMM) [72, 73] is presented in Hansegård *et al.* [74]. The authors demonstrate a method of constraining AAMM to known positions of a number of points. The AAMM is used to learn the statistics of the shape, represented by contours in three planes, along the whole cardiac cycle, preserving the shape-time correspondence. This is achieved by concatenating the shapes in all temporal phases into a single vector. The points to constrain the AAMM are estimated by dynamic-programming-based active contours, and are searched for in the vicinity of the contour provided by the AAMM. Such a fusion of AAMM and active contours in an iterative scheme demonstrates better segmentation accuracy than if these segmentation algorithms were used separately.

Finally, we briefly mention several works on cardiac wall tracking. Orderud *et al.* [75] and Hansegård *et al.* [76] use similar methodology: the cardiac contour is deformed by integrating the search for cardiac boundaries along contour normals into the extended Kalman filter. Essentially, the difference is that in [76] a shape prior based on Principal Component Analysis (PCA) is incorporated into the framework. Another publication from the same group [77] has shown how both epi and endocardium can be coupled in the tracking framework. A classical correlation-based tracking was investigated by Crosby *et al.* [78] and Duan *et al.* [79]. The first one applied normalized correlation to envelope detected beam data directly while a curious feature of the second one is that evaluation has been performed using the open-chest setup (canine hearts).

Among level-set based approaches it is worth to mention a work by Corsi *et al.* [80], who modified the Malladi-Sethian equation for gradient-based image segmentation. They removed the inflationary term to avoid the propagation of the evolving surface beyond regions with missing boundaries (boundary leaking). Since the shape cannot inflate any more, the evolution has to start close to the true boundary. The authors suggested that initialization by placing 5-7 points in 5 short-axis view slices is enough. Three years later a homogeneity-based active contour, integrated into a level-set framework, that does not use image gradient was proposed by Angelini *et al.* [66]. The authors start by denoising the RT3D ultrasound image using brushlets. The idea behind the method is to deform the surface looking for an optimal partitioning of the voxels into homogeneous regions (inside and outside the surface). An interesting fact about the approach is that it extracts highly-curved surfaces with minimal boundary leaking. Another level-set based approach was proposed by Corsaro *et al.* [81]. It is an unconditionally stable 3D semi-implicit

time discretization scheme for solving the level-set formulation of the Riemannian mean curvature flow problem, which allows for fast image segmentation. Using the combination of finite element and finite volume methods they achieved ten times speed-up in comparison to the classical level-sets. Another important contribution is the elimination of the orientation effect (the authors show that the evolved surface that uses left or right oriented triangulation differs from the exact solution and propose their solution to that problem). The proposed framework appears to be robust to vanishing gradients, with a highly curved resulting surface just as in [66]. On the downside there is no evaluation on a database of clinical cardiac images.

Considering the above mentioned methods one can note that many of them use a predefined shape model, which is matched to a 3D image. The benefit of using a predefined shape is that it simplifies establishing links between the model and the cardiac anatomy and allows to easily correlate data between different studies, patients or modalities. It is also easy to subdivide such a model into 17 segments as defined by the American Heart Association (AHA) [82]. The latter will allow to better correlate the results of the algorithm in question to the other algorithms implemented in current echocardiographic systems. From the algorithmic point of view, imposing shape regularity constraints on the predefined model, instantiated to the data, would allow to robustly recover the correct shape even in the areas of ill-defined borders, which are typical for 3DUS. Another interesting problem not addressed by most papers is the segmentation of the epicardium. Being able to segment both epi- and endocardium could provide an interesting insight into myocardial deformation and wall thickening, which is already being measured in other imaging modalities.

In this article we consider the problem of automatic construction of a 3D Active Shape Model (ASM) [7] and using it to segment the epicardium and endocardium of a cardiac left-ventricle (LV) in 3DUS images. Building an ASM would require constructing a model of plausible shape variations (a Point Distribution Model or PDM) and a model of local image appearance in the vicinity of each shape point. These requirements lead to the necessity of having a database of images with delineated myocardial contours. In the imaging modalities which provide high-quality images, such as multidetector computed tomography (MDCT), the process of obtaining shapes, corresponding to the images, can be automated through registration [83], but it is not easy to do in every modality. 3DUS images are rather noisy and have poor level of detail, which might render any registration algorithm ineffective. Therefore we cannot rely only on ultrasound data and propose to automatically construct the PDM from MDCT images as in Ordas *et al.* [83]. The local appearance model, on the other hand, can be automatically learned from synthetic ultrasound data. Having the LV geometry defined by the PDM allows generating a collection of plausible shapes and the corresponding 3DUS images. In this case the cardiac boundaries will be in the positions given by the shape and the size of the set is limited only by the available computational resources. The synthetic data

were generated using a simplified model of ultrasound image formation [84–86] extended to 3D and a more comprehensive approach implemented in Field II [87,88]. The first of these two models, we shall call it FastGen for convenience, assumes a uniform Point Spread Function (PSF), is very fast, and generating one 3D image takes seconds. The second one has a more realistic model of sound propagation, is computationally expensive and requires approximately 30 hours on a single CPU (Intel Xeon 5140, 2.33GHz). In this work we compare the segmentation accuracy of an automatically constructed ASM, trained on real data and on images generated using both ultrasound models.

The paper is structured as follows. We start by a brief description of ASM in Section 4.2, which is used both for providing an ultrasound image generator with the information about cardiac geometry and segmentation of ultrasound images. Section 4.3 introduces the models employed to generate the data for our study. These are used to generate 3DUS images corresponding to given cardiac shapes. Section 4.4 presents the evaluation of the automatically constructed ASM on the sets of synthetic and real images, followed by conclusions in Section 4.6.

4.2 Active Shape Model

The linear Active Shape Model consists of a Point Distribution Model (PDM) and image intensity model. The PDM is constructed by applying PCA to a set of aligned shapes [7] and retaining eigenvectors corresponding to a predefined percentage of shape variability. Shapes are defined by landmark points placed along the contour of the object of interest. The learned shape variability can be modeled by varying $\mathbf{b}$ in the following equation:

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{\Phi}\mathbf{b} \tag{4.1}$$

where $\mathbf{x}$ is an n-vector, representing the shape, obtained by concatenating all the landmark coordinates into a single real-valued vector. In other words, if landmarks have coordinates (x_i, y_i, z_i) the concatenated vector will be of the form $(x_1, y_1, z_1, x_2, y_2, z_2, \ldots)^T$. The n-vector $\bar{\mathbf{x}}$ is the mean of the aligned (by Procrustes analysis) shapes in the training set; the $n \times m$ matrix $\mathbf{\Phi}$ is the eigenvector matrix. Controlling the retained variability (or in other words the number m of retained eigenvectors) controls the level of allowed shape deformation. If the retained variability (usually expressed as percentage of total variability) is low, the model will only provide for the most frequently appearing and strong shape deformations, as learnt from the training set.

The classical approach of matching the model to an image utilizes the profiles perpendicular to the shape at landmark positions. The gradient amplitude of images from the training set of image-shape pairs is sampled along each profile to both sides of the landmark, normalized and used to estimate the mean profile and

covariance matrix. The collection of such pairs for each landmark constitutes an *Intensity Model*. During matching each landmark of the current shape estimate is displaced along the corresponding shape normal as to minimize the Mahalanobis distance between the sampled pixels and the mean profile. The classical approach finds the best position for each landmark among a limited number of candidates N_c: the profile is sampled in N_c equidistant locations along the perpendicular and the best is chosen.

After displacing all the landmarks, the resulting shape is constrained such that its i-th PCA parameter $\mathbf{b}_i$ belongs to the interval determined by the corresponding eigenvalues: $\left[-\beta\sqrt{\lambda_i}, \beta\sqrt{\lambda_i}\right]$. The value of β is typically set to 3 or established experimentally.

The PDM used in our experiments was constructed from a set of high resolution CT images as in [83]. The total training set consisted of 100 MDCT studies of pathologic and asymptomatic patients, 15 temporal cardiac phases each. The ASM intensity model was constructed with 11 samples per profile with $0.5mm$ distance between the samples. The matching was performed using 95% of retained variability, $N_c = 7$ candidates for landmark displacement, and the regularization parameter β equal to 3.

4.3 Generating 3D Ultrasound Images

4.3.1 Fast image generation with FastGen

This ultrasound image generation method follows an approach originally proposed by Bamber and Dickinson [86]. It is assumed that the imaging system can be modeled by a linear, space-invariant point spread function (PSF). Let $t(x, y, z)$ be an echogenicity model (an image with different intensity values corresponding to different tissues, see Fig. 4.1b) of the object being imaged (Fig. 4.1a). The x, y and z are lateral, elevation and axial coordinates. First, scatterer distribution is modeled by multiplying the echogenicity model by a Gaussian white noise $G(\sigma_n; x, y, z)$ with zero mean and variance σ_n^2 (Fig. 4.1c):

$$T(x, y, z) = t(x, y, z) \cdot G(\sigma_n; x, y, z) \tag{4.2}$$

The 3D ultrasonic echo dataset $V(x, y, z)$ can then be obtained by a convolution

$$V(x, y, z) = h(x, y, z) * T(x, y, z) \tag{4.3}$$

where

$$h(x, y, z) = h_1(x, \sigma_x) \cdot h_1(y, \sigma_y) \cdot h_2(z, \sigma_z) \tag{4.4}$$

$$h_1(u, \sigma_u) = exp\left[-u^2 / \left(2\sigma_u^2\right)\right] \tag{4.5}$$

$$h_2(v, \sigma_v) = sin(2\pi f_0 v/c)\, exp\left[-v^2 / \left(2\sigma_v^2\right)\right] \tag{4.6}$$

c is the speed of sound in a soft tissue (assumed 1540 m/s) and f_0 is the center frequency of the transducer. $f_0 = 3$MHz is used throughout this article, being a typical frequency for cardiac imaging.

The image of the envelope-detected amplitude, $A(x, y, z)$ (shown in Fig. 4.1d), is given by

$$A(x, y, z) = \left| V(x, y, z) + i\widehat{V}(x, y, z) \right| \tag{4.7}$$

where $\widehat{V}(x, y, z)$ is the Hilbert transform of $V(x, y, z)$ and i is the imaginary unit.

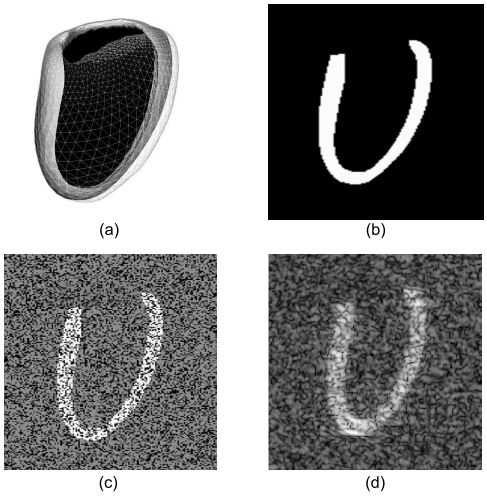


Figure 4.1: Ultrasound image generation by FastGen. The shape (a) is used to generate an echogenicity model (b) where different voxel intensities represent different tissues. Subsequently Gaussian noise is added to introduce scatterer variations (c), the result is convolved with the PSF (4.4) and the envelope of the convolution result is computed (d).

4.3.2 Image generation using Field II

This approach relies on linear systems theory to find the ultrasound field for the cases of both pulsed and continuous wave. The latter is performed using the spatial impulse response, which, when the transducer is excited by a stimulus, modeled by the Dirac delta function, gives the emitted ultrasound field at a specific point in space as a function of time. The field for any kind of excitation can then be found by convolving the spatial impulse response with the excitation function. The modeling program allows for any transducer geometry (unlike that from the previous subsection) and apodization, and uses a far-field approximation, to keep the process realistic and simple. The approach assumes a homogeneous bounded medium, where the pressure is sufficiently small to ensure linear wave propagation.

The method can be described using Huygens' principle, where the impulse response is calculated from a summation of all spherical waves from the aperture area

S as:

$$h\left(\mathbf{r}_1, t\right) = \int_S \frac{\delta\left(t - \left|\mathbf{r}_1 - \mathbf{r}_2\right|/c\right)}{2\pi\left|\mathbf{r}_1 - \mathbf{r}_2\right|} dS \tag{4.8}$$

where $|\mathbf{r}_1 - \mathbf{r}_2|$ is the distance from the transducer at position $\mathbf{r}_2$ to the field point $\mathbf{r}_1$, $\delta(t)$ is the Dirac delta function, and c is the speed of sound. For a number of apertures (like round piston, circular convex element, rectangular element) the calculation can be done analytically. However it is not possible for a general aperture. Field II therefore divides the aperture into smaller mathematical elements to describe advanced shapes. Subsequently, any kind of linear ultrasound field can be calculated using spatial impulse responses. The emitted pressure field $p(\mathbf{r}_1, t)$ is given by

$$p\left(\mathbf{r}_1, t\right) = \rho_0 \frac{\partial v\left(t\right)}{\partial t} * h\left(\mathbf{r}_1, t\right) \tag{4.9}$$

where ρ_0 is the density of the medium and $\partial v(t)/\partial t$ is the acceleration of the front face of the transducer. The received voltage signal for the pulse echo field is:

$$v_r\left(\mathbf{r}_1, t\right) = v_{pe}\left(t\right) * f_m\left(\mathbf{r}_1\right) * h_{pe}\left(\mathbf{r}_1, t\right) \tag{4.10}$$

$$f_m\left(\mathbf{r}\right) = \frac{\Delta\rho\left(\mathbf{r}_1\right)}{\rho_0} - \frac{2\Delta c\left(\mathbf{r}_1\right)}{c} \tag{4.11}$$

where the scattering signal $f_m(\mathbf{r})$ arises from spatial variations in density $\Delta\rho(\mathbf{r}_1)$ and speed of sound $\Delta c(\mathbf{r}_1)$. Here $h_{pe}(\mathbf{r}_1, t)$ is the two-way spatial impulse response, which is a convolution between the impulse response of the transmitting and receiving aperture. The impulse response $v_{pe}(t)$ includes the excitation convolved with the transducer's electro-mechanical impulse response in both transmit and receive. For further details, please, refer to Jensen *et al.* [87,88].

The image generation process is similar to that of FastGen (see Fig. 4.2) with an exception that the echogenicity model is not defined on a uniform grid anymore. It is created by populating the 3D space with randomly positioned point scatterers, each having an amplitude that characterizes the strength of the reflected signal. Subsequently, Field II uses this echogenicity model together with a custom transducer definition to obtain the ultrasound image.

4.4 Evaluation datasets

4.4.1 Synthetic Training and Testing Sets

The FastGen generated training set for the intensity model consisted of 270 volumes. The following parameters were chosen to introduce variability in both shape

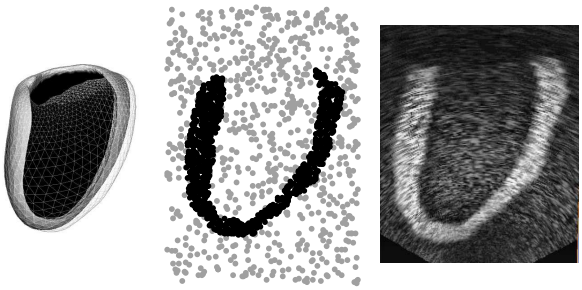


Figure 4.2: Ultrasound image generation using Field II. The shape (a) is used to randomly populate a 3D space with point scatterers (b), each of them having an assigned value that characterizes strength of response. The image is obtained by modeling sound wave propagation (c). The black points in (b) correspond to myocardium. Note that different densities of points have been used only for better visualization.

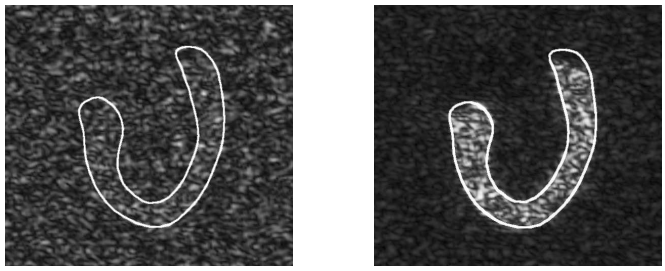
Table 4.1: Model parameters for FastGen.

Parameter	**Training Set**	**Testing Set**
σ_z [mm]	0.30, 0.50, 0.70	0.30
σ_x, σ_y [mm]	0.50, 0.75, 1.00	0.70
σ_n	0.75, 1.00, 1.25	0.70
Myocardium intensity	250	60, 75, 90, . . ., 255
Blood pool intensity	60	40
Background intensity	70	50

Table 4.2: Model parameters for Field II.

Parameter	**Training Set**	Testing Set
Number of scanlines	40, 60, 80	80
Active elements	128, 256, 512	512
Number of scatterers ($\times 10^3$)	500, 1000, 1500, 2000	2000
Myocardium intensity	250	60, 75, 90, ..., 255
Blood pool intensity	60	40
Background intensity	70	50
Probe size [mm]	20×15	
Field of view [°]	120	
Speed of sound [m/s]	1540	
Center frequency [MHz]	3	
Sampling frequency [MHz]	100	
Focus depth [mm]	70	
Probe matrix dimensions [mm]	15×20	
Depth of cardiac apex [mm]	20	
Piezoelement matrix [elements][1]	32×64	

[1] Dimensions of each element were computed as the physical size of the probe divided by the number of elements in rows and columns. The distance between elements was computed as dimensions of an element divided by 1000. Most of the parameters are the same as in the examples coming with Field II.



(a) (b)

Figure 4.3: Sample images, created by FastGen, corresponding to the low (a) and high (b) intensity differences with superimposed shapes, which were used to generate these images.

geometry and speckle appearance pattern. Ten shapes corresponding to the random variations (within $\pm 1.5\sqrt{\lambda}$) of the first five principal components (PC) of the PDM were generated. The SD of Gaussian envelope over the sine wave in axial direction σ_z was taken equal to $0.30mm$, $0.50mm$, $0.70mm$. The SDs of Gaussian envelope over sine wave in lateral and elevation directions (σ_x and σ_y) were taken equal $0.50mm$, $0.75mm$, $1.00mm$. The SD of the Gaussian noise σ_n was taken equal 0.75, 1.00, 1.25. In order for the training set to have only noise pattern variation the intensities were kept constant. Their values have been chosen empirically to have good visual contrast between tissues and are equal to 270 for the myocardium, 70 for the background and 60 for the blood pool. Due to the profile normalization, the model trained on these data should be able to deal with images that have different tissue contrast.

The testing was performed on both synthetic and real data. The synthetic testing set was generated with an idea of providing images of different tissue contrast. It consisted of 280 volumes. For generating the synthetic images 20 shapes corresponding to the random variations (within $\pm 1.5\sqrt{\lambda}$) of the first five PCs of the PDM were generated. Noise variances have been chosen empirically to reproduce the real images as much as possible. They are: $\sigma_z = 0.30mm$ for axial direction, $\sigma_x = \sigma_y = 0.70mm$ for lateral and elevation, $\sigma_n = 1.20mm$ for the Gaussian noise. Intensities $t(x, y, z)$ (from 0 to 255) are $60, 75, \ldots, 255$ for myocardium; 50 for background (40 for the blood pool). So the myocardium contrast (difference in intensities between the myocardium and the background) varies from 10 to 205 in steps of 15. Two sample images can be seen in Fig. 4.3.

The generation of the training and testing sets with Field II followed similar guidelines, taking into account its specifics. Since there were more parameters to tune, the training set was slightly larger and consisted of 360 images while the testing again contained 260 images. All the parameters are summarized in Tables 4.1

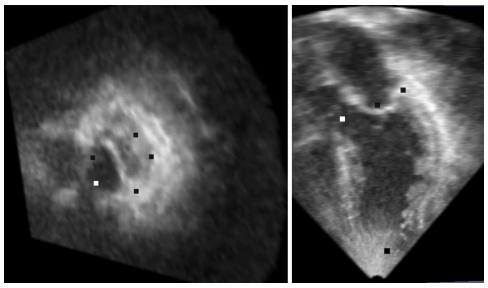


Figure 4.4: Short axis and long axis views of the points used for model initialization superimposed on an image. The white point corresponds to the aortic valve and provides an orientation cue.

and 4.2.

The initial shape for the segmentation of synthetic images was the mean shape of the PDM (consisting of 2677 points) aligned to the ground-truth shape by a similarity transform.

4.4.2 In-vivo Training and Testing Sets

The *in-vivo* set consisted of manually landmarked end-systole(ES) and end-diastole(ED) LV 3D volumes of 20 cardiac resynchronization therapy (CRT) patients. The data were acquired using a Philips IE33 echograph (Philips Ultrasound Inc., Andover, USA) with X3-1 transducer and exported into accessible format using Philips QLAB v6.0 quantification software. The exported data are envelope detected $224 \times 208 \times 208$ images with voxel size approximately $1.0 \times 1.0 \times 0.7$mm. Sample images can be seen in Fig. 4.5.

Due to the small quantity of data, the experiments that involved *in-vivo* training set used a leave-one-out cross validation (all the images of the same patient had been excluded).

The initial shape for the segmentation of *in-vivo* images was the mean shape aligned by an affine transform to six points: four points on the endocardium in the basal plane, the center of the aortic valve, and the apex, as illustrated in Fig. 4.4.

4.5 Experiments

4.5.1 Validation on synthetic data

To start, we wanted to verify that ASM trained on the synthetic data can successfully segment the synthetic data of different contrast produced by the same algorithm. For this purpose FastGen and Field II generated testing sets have been segmented by the ASM trained on the corresponding training sets. The results are shown in Fig. 4.6. The segmentation algorithm is not completely invariant to the image contrast variation, as it could be expected due to intensity normalization, and the

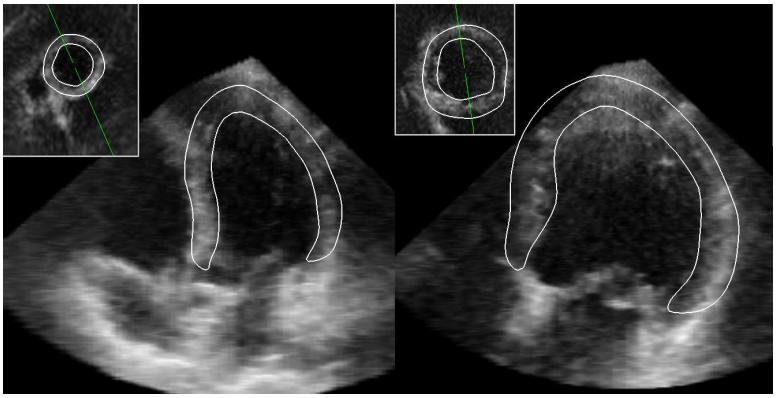


Figure 4.5: Real ultrasound images (from 2 patients) from our testing set with superimposed manual delineations.

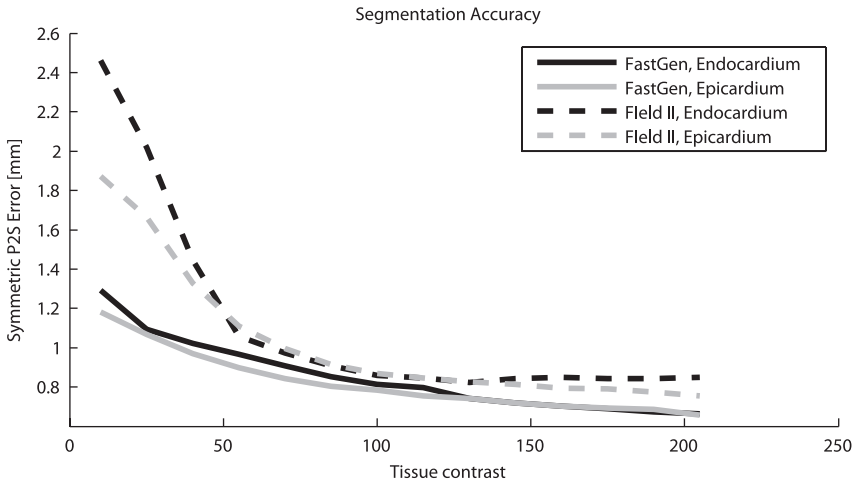


Figure 4.6: Accuracy of segmenting the simulated images with varying tissue contrast. The contrast is expressed as difference of scatterer amplitudes. Shown are the symmetric P2S errors for endocardium and epicardium.

images with higher tissue contrast have higher segmentation accuracy. Nevertheless the ASM trained on Field II set appears to have slightly inferior performance on the low-contrast data.

The symmetrical point-to-surface error is defined for two shapes (surface meshes) s_1 and s_2 as:

$$\varepsilon\left(s_1, s_2\right) = \left[d\left(s_1, s_2\right) + d\left(s_2, s_1\right)\right] / 2 \tag{4.12}$$

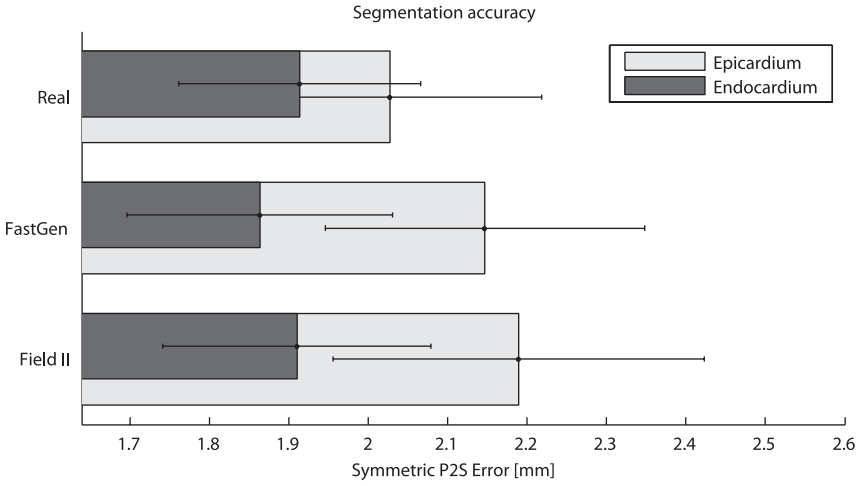
where

$$d\left(s_1, s_2\right) = \frac{1}{N} \sum_{i=1}^{N} \left\| s_1\left(i\right) - \arg\min_{p \in s_2} \left\| s_1\left(i\right) - p \right\|^2 \right\| \tag{4.13}$$

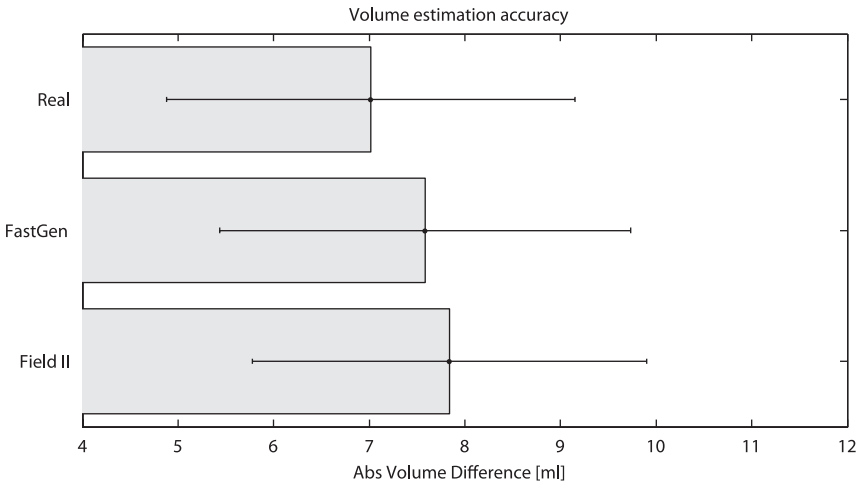
is an asymmetric point-to-surface error, which is the mean distance between each vertex of the mesh s_1 and mesh s_2 ($s_1(i)$ refers to the i-th vertex of the mesh s_1); $\|\cdot\|$ is the l^2 norm.

4.5.2 Evaluation on real images

In this section we compare the ASM trained on real images and artificially generated images and see whether there is any benefit of using the latter. In the case when real images had been used for training a leave-one-out strategy was employed (training on 38 images, corresponding to 19 patients, and segmenting the ED and ES of the remaining patient). The segmentation accuracy is shown in Fig. 4.7. The results of the volume and ejection fraction (EF) estimation by the best approach (the ASM trained on FastGen data) are summarized in Table 4.3. The accuracy was measured with respect to manually delineated contours. As one can note from the confidence intervals, the difference between the different training sets is not statistically significant (with 95% confidence level) although there seem to be an improvement when FastGen is used. There are of course several issues that are worth commenting, assuming that all the approaches perform equally well. It would be expected for the Field II to be superior to FastGen due to being a more realistic model. Nevertheless Field II has many more parameters to tune and generating a representative training set would require sweeping over all of them (when the parameters of the equipment are unknown, which is the usual case). The latter would lead to an enormous database, but Field II in 3D is very computationally demanding. It takes about half an hour to generate one 3D volume on a 60-processor cluster fully dedicated to the task. Theoretically this would imply one week for a set of 280 volumes. Using real data for ASM training has its own downside in that it requires a lot of manual delineating in 3D, which is very time consuming. It is also very difficult to produce consistent delineations across different phases (like ED and ES) mostly due to the noisiness and rather poor quality of the images. FastGen is free from both of these disadvantages.



(a)



(b)

Figure 4.7: Accuracy of segmenting the real images by an ASM trained on different training sets in terms of (a) the mean symmetric point-to-surface error for endocardium and epicardium and (b) mean absolute volume difference for LV cavity. Error bars represent 95% confidence interval of the mean.

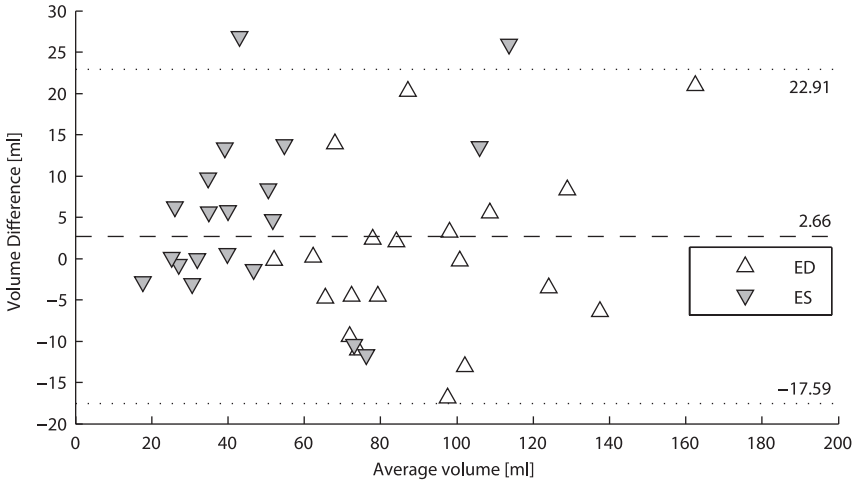


Figure 4.8: Bland-Altman plot of the volume estimation accuracy on the real images by an ASM trained on FastGen training set.

The Bland-Altman and scatter plots for the best approach (FastGen) are shown in Figs. 4.8, 4.9 and some segmentation examples can be seen in Fig. 4.10. The scatter plot suggests a nice linear relationship with high correlation between the measurements and ground truths. The slope of the fitted line is smaller than 1.0 and intercept greater than 0.0 as is expected when there is no relation between the error and magnitude [89].

An interesting question that can be answered using the image generation is how much data in the training set is actually needed for accurate segmentation. The size of the training sets in the experiments has been chosen from practical considerations, trying to make the Field II training set in a reasonable time. But since FastGen is really fast, we can use it to increase the size of the training set and see how it affects the segmentation accuracy. To generate larger training set with FastGen we used the same parameters as before but with more shapes (40 different shapes in total). The biggest training set consisted of 1080 image-shape pairs. 36 training sets have been generated: starting from 4 random shapes, all variations of all the parameters (27 images per shape), and randomly adding one shape per training set.

The results can be seen in Fig. 4.11. As expected, the error on the artificial testing set is decreasing with the size of the training set and stabilizes around 1000, but with acceptable results already around 500. Of course these numbers are bound to the training and testing sets used and after 1000 images the training fails to offer more information to the intensity model. On the other hand the training set size does not seem to have much effect on real data, although very wide confidence in-

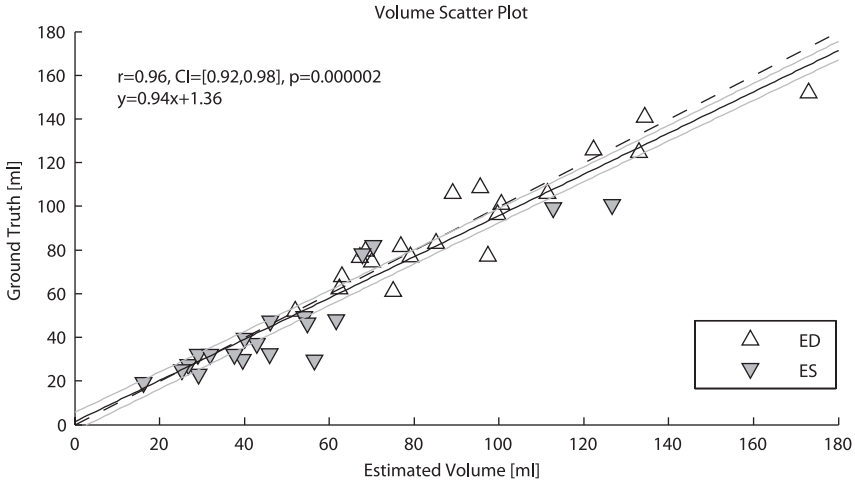


Figure 4.9: Comparison of volumes estimated by the proposed algorithm and ground truths. The solid black line is the fitted one, dashed line is the equality line, gray lines represent a 95% confidence interval of the fit. The CI is the 95% confidence interval of the correlation coefficient.

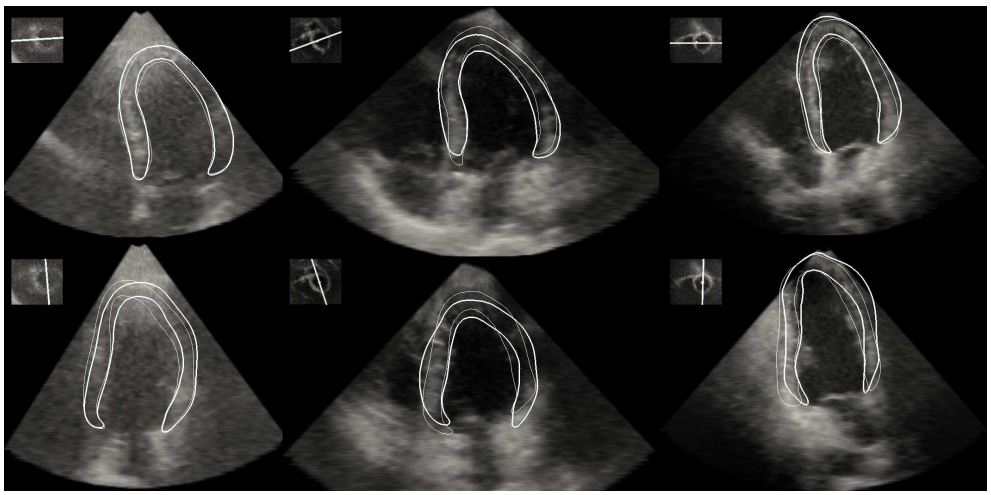


Figure 4.10: Segmentation examples. Shown are three hearts, one per column, two perpendicular views per heart. The cut planes chosen are shown in the thumbnail in the upper left corner and are aligned with the long axis. Thin line represents manual delineation while thick - automatic one, by the ASM trained on FastGen data.

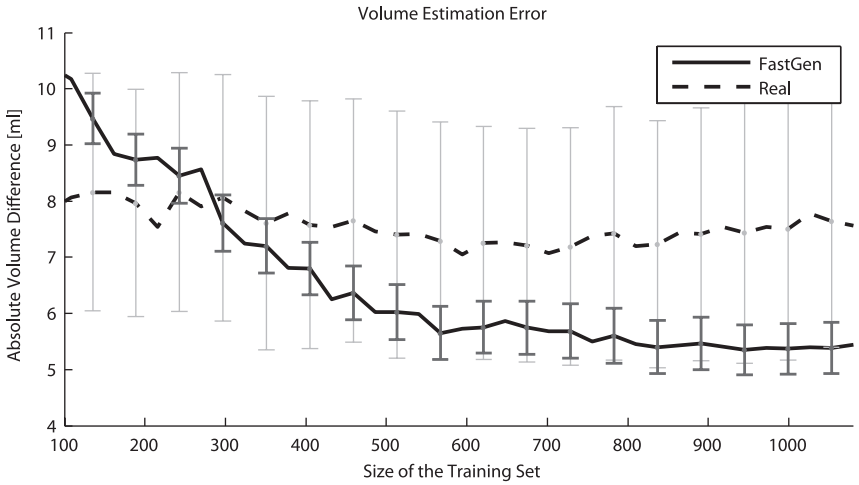


Figure 4.11: Segmentation accuracy measured in terms of mean absolute volume difference for different sizes of the FastGen training set evaluated on the real and FastGen generated test data. Error bars correspond to 95% confidence interval of the mean.

terval hampers drawing any conclusions from the plot. The possible reason could lie in the combination of insufficient image quality (which in general results in high interobserver variability and our results are not far from it) and simplified ultrasound model of FastGen. Nevertheless, although there is no evidence to support the use of more training data, the computational simplicity of the image generation algorithm allows to easily obtain large quantity of training data and therefore the segmentation algorithm can easily benefit from it.

Table 4.3: Evaluation results on the real datasets by an ASM trained on FastGen data. Together with the measurements 95% confidence intervals are shown.

	LV Volume [ml]	**LV EF [%]**
Mean Difference	2.66 ± 2.19	-5.84 ± 2.93
Limits of Agreement	$\pm 20.24 (\pm 5.54)$	$\pm 18.55 (\pm 2.59)$
RMSE	10.54	10.92

P2S Error [mm]	
Epicardium	2.15 ± 0.20
Endocardium	1.87 ± 0.16

4.5.3 Comparison to other methods

To put our work in the context of other contemporary papers, Tables 4.4 and 4.5 present the results of several techniques recently reported in the literature. Shown is the mean with 95% limits of agreement ($mean \pm t \cdot SD$, where t is taken from the Student's t Table, whenever authors of these articles reported the accuracy as $mean \pm SD$ we multiplied SD by the corresponding t). The value in the column `Volume` means that the authors did not classify volumes into EDV/ESV. An exception is the work by Angelini *et al.* [66], who provided both estimated and ground truth volumes, which allowed us to compute the values we needed. In the latter case in the column `Volume` we merged the results for ESV and EDV in order to compare the result with other papers. It can be seen from the table that our results are consistent with those of the state-of-the-art algorithms (considering accuracy of EF estimation) and better in estimating the volumes themselves (on average). Nevertheless, in terms of EF, the results are still much worse than those corresponding to the intra- and interobserver variability reported in other studies (See Table 4.6). On the other hand it is interesting to note that the accuracy of volume estimation reported by other clinical studies is also on the same level [90], for instance Jenkins *et al.* [91] reported the scatter of EDV at the level of $\pm$29ml and ESV - $\pm$18ml for 50 patients (although the measurements were compared to those in MR and the error could accumulate) nevertheless the EF was significantly smaller - 0 ± 7% which indicates consistency in inaccuracies between ED and ES. In our case the scatter of EF was rather large and the consistency has to be improved.

Let us go over all the approaches presented in Table 4.4 and start with the constrained AAMM fused with active contours and unconstrained AAMM of Hansegård *et al.* [74]. The major disadvantage of the approach is that AAMM represents both the shape deformation at a certain time instant and the motion pattern. In order to build a representative model, the training set should have not only all the possible cardiac shapes but all the possible deformation patterns as well. This calls for a large training set with manual delineations in every temporal phase, which is difficult to obtain. An automatic generation of such a training set using an ultrasound image generator combined with realistic LV deformations, extracted from MDCT data, or using a biomechanical model, would really benefit the approach. On the other hand, the way the constraints are imposed on the points requires to know the point correspondence and combining AAMM with Active Contours requires tuning both their parameters plus a coupling weight. On the positive side, AAMM allows for realistic modeling of both shape and cardiac deformation (and texture of course if needed).

The level-set based segmentation, that partitions the image into homogeneous regions without advection term and gradient information of Angelini *et al.* [66]. The approach depends on spatiotemporal brushlet denoising which adds additional parameters to tune to the level-set specific parameters. Since there is no prior,

leaking is possible and the resulting boundary lacks smoothness. The advantage is that the algorithm does not require any training set and can be initialized anywhere in the image (the algorithm tracks only one closed blob centered around the center of mass of the object of interest).

The wiremesh of Zagrodsky *et al.* [70] employs 3D Sobel edge detector which relies on Median filtering. Although these two filters do not require any significant tuning, the median filter is probably not the best choice for ultrasound images, and many spurious edges might still remain on the image. The algorithm requires balancing internal and external forces and the execution speed is very low: 3 min for registration, 8 min for segmentation on a dual 1.7GHz Pentium. On the other hand the approach does not require a training set and the initialization is automated through registration.

Finally, a modified Malladi-Sethian equation to avoid leaking of Corsi *et al.* [80] is rather a typical level-set segmentation. It is automatic and no training set is required. It does not use any shape prior and leaking is prevented essentially by removing the advection term.

What we have proposed is an approach that uses explicit shape representation with automatically constructed models of the shape and local appearance. If we compare it to the above approaches there is no limitation on the complexity of the shape; user interaction is reduced only to model initialization during segmentation and tuning of ASM parameters (which can usually be left unchanged). For example, introducing epicardium into segmentation pipeline is much easier than in the above approaches, except for the AAMM (it would only require more work on manual delineations). It is done simply by merging meshes of both structures.

Another advantage of our approach lies in using multi-modal data. The shape model, automatically created from a large amount of MDCT images, provides an accurate shape model as opposed to the one that might have been constructed from ultrasound images, where the boundaries are poorly defined and have to be guessed. The generation of artificial ultrasound images, on the other hand, avoids having incorrect LV delineations in real data.

4.6 Conclusions

In this paper we proposed an approach to automatically construct an ASM for 3DUS segmentation. In contrast to the majority of 3DUS segmentation algorithms, our model includes both epi- and endocardium, providing simultaneous segmentation of a complete left ventricle. The approach is based on the combination of automatically constructed shape model from MDCT and local appearance learned from data sets automatically generated using two commonly accepted models of ultrasound. One of them, FastGen, is fast and assumes uniform PSF, while the other, Field II, has more realistic, but linear, sound propagation model. It has been shown

Table 4.4: Summary of LV segmentation algorithms.

Ref	Algorithm	Year	Modality	Dataset	Equipment	Gold Standard
	Our approach, automatically constructed ASM.		RT3D	20 CRT patients	Philips IE33 echograph with X3-1 transducer	Manually traced contours in RT3D
[74]	DP-CAAMM, constrained AAMM fused with active contours	2007	Triplane US	36 adults (old myocardial infarction, valve disease, DCM, pulmonary hypertension, heart transplant, structurally normal hearts)	GE Vivid 7 with matrix-phased array transducer 3V	Manual landmarks in RT3D
[74]	Unconstrained AAMM	2007	Triplane US			
[66]	Homogeneity-based active contour	2005	RT3D	10 patients with pulmonary hypertension: 8 with primary PH, 2 secondary PH associated with congenital defects		Manually traced contours in RT3D
[70]	Wiremesh	2005	RT3D	10 patients (healthy and diseased)	Volumetrics and Philips SONOS 7500	Manually traced contours in RT3D
[80]	Modified Malladi-Sethian equation	2002	RT3D	20 patients	Volumetrics	Manual volume measurements in MRI

Table 4.5: Accuracy of the LV segmentation algorithms. The errors are given as $mean \pm t \cdot SD$ with the 95% confidence of the mean in the parenthesis. t was chosen according to the number of images to provide 95% limits of agreement.

Ref	Algorithm	Year	EDV [ml]	ESV [ml]	EF [%]	Volume [ml]
	Our approach, automatically constructed ASM.		0.1 ± 21.1 (± 4.7)	5.2 ± 21.1 (± 4.7)	-5.8 ± 18.5 (± 2.9)	2.6 ± 20.2
[74]	DP-CAAMM, constrained AAMM fused with active contours	2007	-3.1 ± 40.6 (± 6.8)	0.61 ± 26.4 (± 4.4)	-1.3 ± 12.8 (± 2.1)	n/a
[74]	Unconstrained AAMM	2007	-7.3 ± 40.6 (± 6.8)	-2.5 ± 44.7 (± 7.4)	-1.5 ± 22.3 (± 3.7)	n/a
[66]	Homogeneity-based active contour	2005	16.1 ± 57.8 (± 18.3)	6.6 ± 39.7 (± 12.5)	0.6 ± 25.6 (± 8.1)	11.4 ± 45.8
[70]	Wiremesh	2005	-0.1 ± 49.3 (± 15.6)	-4.2 ± 32.3 (± 10.2)	2.6 ± 21.1 (± 6.7)	-2.1 ± 37.8
[80]	Modified Malladi-Sethian equation	2002	n/a	n/a	n/a	-15.6 ± 41.1

Table 4.6: Intra- and interobserver variabilites as reported in other studies.

Ref	Authors	Year	EDV	ESV	EF
	Interobserver Variability				
[74]	Hansegård *et al.*	2007	13.0 ± 38.6ml	9.9 ± 30.5ml	$-1.7 \pm 12.8\%$
[92]	Sugeng *et al.*	2006	$11.2 \pm 17.6\%$	$14.2 \pm 24.1\%$	$10.5 \pm 17.0\%$
	Sugeng *et al.*, adapted to our data		10.3 ± 15.6ml	13.1 ± 21.4ml	$5.5 \pm 8.5\%$
	Intraobserver Variability				
[92]	Sugeng *et al.*	2006	$3.9 \pm 4.0\%$	$5.6 \pm 8.0\%$	$5.6 \pm 6.9\%$
	Sugeng *et al.*, adapted to our data		3.6 ± 3.6ml	2.5 ± 3.4ml	$2.9 \pm 3.5\%$

that although using synthetic images to train an ASM demonstrates a similar segmentation accuracy as training ASM from manually delineated images, the former are much faster to obtain if FastGen is used. It is also much easier to control the image quality of the generated data, while with real data the quality depends a lot on the patient and many of them do not have a good acquisition window.

The best segmentation results were obtained by FastGen, resulting in the volume estimation accuracy with limits of agreement across the whole population of 2.6 ± 20.2ml. The average point-to-surface segmentation errors for epicardium and endocardium were 2.15 ± 0.20mm and 1.87 ± 0.16mm, respectively.

The synthetic 3DUS images have two clear benefits over the real ones: high quality and detail and an already solved correspondence problem between a shape and an image. The latter avoids any manual landmarking, which is error prone, complicated in 3D and shows high interobserver variability (approximately 10.3 ± 15.6ml for EDV and 13.1 ± 21.4ml for ESV) [92]. On the other hand, it is very difficult to obtain a large set of good quality real 3DUS images, while by generating them artificially we can obtain a set of any size. The choice of the ultrasound propagation model is dependent on the segmentation algorithm. In the case of the classical linear ASM, the advanced model of Field II does not seem to improve the intensity model and FastGen can be beneficial in terms of time, computational resources and implementation effort.

The high computational cost of Field II and lack of accurate information about the employed ultrasound probe did not allow us to fully investigate the construction of an optimal synthetic training set. Still, the biggest drawback of the proposed methodology is its lack of consistency when segmenting the data of the same patient (as can be observed from EF errors) and it would benefit from borrowing some ideas from the tracking algorithms such as adaptation to the observed data. Automating the initialization would also be convenient. Currently we have seen automatic initialization based on registration and Hough transform. Registration, though, takes too much time and is unreliable in ultrasound. Hough transform, on the other hand, combined with filtering and edge detection could be a viable approach as shown by Stralen *et al.* [93].

CHAPTER 5

Automatic Construction of 3D-ASM Intensity Models by Simulating Image Acquisition: Application to Myocardial Gated SPECT Studies

Abstract - *Active shape models bear a great promise for model-based medical image analysis. Their practical use, though, is undermined due to the need to train them on large image databases. Automatic building of Point Distribution Models (PDMs) has been already successfully addressed in the literature. However, the need for strategies to automatically build intensity models has been largely overlooked. This work demonstrates the potential of creating intensity models automatically by simulating image generation. We show that it is possible to reuse a 3D PDM built from Computed Tomography (CT) to segment gated Single Photon Emission Computed Tomography (gSPECT) studies. Training is performed on a realistic virtual population where image acquisition and formation have been modeled using the SIMIND Monte Carlo simulator and ASPIRE image reconstruction software, respectively. The dataset comprised 208 digital phantoms (4D-NCAT) and 20 clinical studies. The evaluation is accomplished by comparing point-to-surface and volume errors, against a proper gold standard. Results show that gSPECT studies can be successfully segmented by models trained under this scheme with sub-voxel accuracy.*

Adapted from C. Tobon-Gomez, C. Butakoff, S. Aguade, F.M. Sukno, G. Moragas, A.F. Frangi. Automatic Construction of 3D-ASM Intensity Models by Simulating Image Acquisition: Application to Myocardial Gated SPECT Studies. *IEEE Transactions on Medical Imaging*, 27(11):1655–1667, 2008.

5.1 Introduction

In spite of the high technological developments in medical imaging systems for diagnostic cardiology, cardiac function is still mostly analyzed through visual assessment or manual delineation, which are both time consuming, subjective and error prone. This fact has generated the need for automated analysis tools to support diagnosis with reliable and reproducible image interpretation. However, the success of currently available commercial packages is modest and their use underdiffused.

On the one hand, automated delineation of the cardiac chambers from 3D and 4D image datasets is challenging. Recent surveys have pointed out the prevalence of model-based approaches to accomplish this task [65, 94]. Typically, they require a generic template which undergoes adaptation to fit specific image data. This strategy enables introducing *a priori* knowledge of shape of the structure of interest into the segmentation process. In particular, Active Shape Models (ASMs) [6] have been successfully employed in image segmentation [21, 95]. Unfortunately, construction of these models requires several training steps based on a target image database (ideally a rather extensive one). This is simply unachievable by sole manual processing on 4D datasets due to the huge amount of data involved. These steps include: *i*) manual outlining of target boundaries, *ii*) consistent distribution of landmarks across sample shapes, *iii*) statistical shape decomposition yielding a Point Distribution Model (PDM) [6], and *iv*) learning a statistical model of the intensity around the target object. Substantial efforts have been carried out to automatically construct PDMs by autolandmarking surface [32,96,97] or volumetric [98,99] representations of already segmented structures. Some authors have shown techniques which circumvent the need for segmenting all sample volumes and work directly from the raw images [83, 100].

To the best of our knowledge, no work has attempted to automate the process of creating intensity models. This is precisely the focus of this work, which we use to complement our fully automatic ASM construction strategy initiated with the autolandmarking method by Frangi *et al.* [98], and more recently by Ordas *et al.* [83]. We show that it is possible to build a 3D-ASM, suitable for segmentation of gated Single Photon Emission Computed Tomography (gSPECT) images, with a PDM previously built from a large database of cardiac Computed Tomography (CT) data [83]. The use of a virtual population provided access to known LV surfaces for training purposes and accuracy evaluation.

On the other hand, imaging simulators are currently a mature field of research providing tools for a variety of modalities: SPECT [101–103], CT [104, 105], Ultrasound (US) [88], and Magnetic Resonance Imaging (MRI) [106, 107]. Among them, SPECT simulators have the longest trajectory, hence they now offer straightforward tools for cardiac applications. This has motivated the use of gSPECT as a show case for the usage of our approach. Nonetheless, the underlying concepts regarding

automatic building of statistical models can be applied to other major diagnostic imaging modalities.

Segmentation of the LV cavity from SPECT imaging is a challenging problem owing to limitations inherent to the modality (i.e. low resolution, blurred boundaries, high noise levels, signal drops, absence of anatomical landmarks, etc) [108]. Model-based post processing algorithms are quite widespread in clinical practice [109, 110]. Yet their quantifications are affected by intrinsic imaging drawbacks, specially in patients with small or hypertrophic hearts [111]. Similarly, less accurate calculations have been found in the presence of extracardiac activity, low-dose studies or severe perfusion defects [112, 113]. Hence, new approaches able to cope with these constraints are highly desirable. Deformable models [114] and level set based [115] algorithms are more sophisticated approaches previously applied to SPECT segmentation, giving promising results on simulated data. Still, further validation on real clinical cases is needed.

This manuscript is organized as follows: The theoretical background of ASMs is explained in Section 5.2. The datasets used for our experiments are presented in Section 5.3. A detailed description of the methodology for automatic construction of 3D-ASM intensity models is provided in Section 5.4. Section 5.5 presents the experimental setup of this work, followed by its results in Section 5.6. Section 5.7 aims to further discuss the obtained results. Finally, the last section exposes the clinical contribution and outlook of our work.

5.2 Background

A concise explanation of Active Shape Models (ASM) is provided in the current section. An extended description can be found in [6].

Basically, three main parts constitute the backbone of an ASM: A shape model, an intensity model, and a matching algorithm. The shape model (PDM) represents the shape variability of the object under study. For a three dimensional space, a linear PDM constructed from n aligned shapes, $\{\mathbf{x}_i; i = 1, \ldots, n\}$, of m landmarks each, $\{\mathbf{l}_j = (l_{xj}, l_{yj}, l_{zj}); j = 1, \ldots, m\}$, is a linear model defined by:

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{\Phi}\mathbf{b} \tag{5.1}$$

where $\mathbf{x}$ is a $3m$-element vector obtained by concatenating all landmark coordinates in the form $(l_{x1}, l_{y1}, l_{z1}, l_{x2}, l_{y2}, l_{z2}, \cdots, l_{xm}, l_{ym}, l_{zm})$. Then, $\bar{\mathbf{x}}$ is the mean of aligned shapes in the training set, $\mathbf{b}$ is the shape parameter vector of the model, and $\mathbf{\Phi}$ is a matrix whose columns are the principal components of the covariance matrix:

$$\mathbf{S} = \frac{1}{n-1} \sum_{i=1}^{n} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \tag{5.2}$$

Obtaining the m 3D landmarks and their correspondence for all points on every dataset is not a trivial task. Our methodology was inspired on the method proposed by Frangi *et al.* [98]. Because of our particular application, a one chamber model (LV) was used. Such configuration is a subpart of our recently constructed *whole heart model*, trained from a high-resolution CT dataset [83]. Its training included 100 subjects in 15 temporal phases. Thus, 1500 sample volumes were considered in total.

Once the shape model has been established, the second component (intensity model) comes into action. It aims to grasp the intensity distribution typically found near the object's boundaries. It does so by sampling the gradient of the intensity profiles along the perpendiculars to the mesh. From pixels sampled along each profile, the mean vector and covariance matrix are estimated and stored for later use during matching. An intensity model was calculated for each endocardial and epicardial wall of the 17 LV AHA's segments [82]. Hence a total of 33 regions were obtained, corresponding to 17 epicardial and 16 endocardial.

Finally, the third element (matching algorithm) has the role of deforming the mesh to match image data. Our approach is based on the sparse fitting method, SPASM, put forward by van Assen *et al.* [21]. We modified this technique by using an intensity model where each candidate point is obtained by selecting the minimal Mahalanobis distance between the sampled profiles and the mean profiles of the intensity model. Candidate points operate as deformation forces propagated to neighboring nodes with a weight function

$$w(\lambda, \omega) = e^{-\frac{\|\lambda-\omega\|^2}{2\sigma_p^2}} \tag{5.3}$$

where $(\|\lambda - \omega\|^2)$ is the geodesic distance between nodes, and σ_p is the width of the normalizing Gaussian kernel. Deformation forces drive the mesh to a best-fit location after several iterations. The steps of the algorithm are illustrated in **Algorithm 2**.

5.3 Materials

Two main datasets were used for this work: a virtual and a clinical population. The virtual population consisted of digital phantoms (see Section 5.4.1 for details) and was considered for 3D-ASM intensity model training. Afterwards, it was employed to evaluate performance of the trained models by means of *leave-one-out* approach: Each case was segmented by a model trained with all cases but itself (in total $n - 1$ cases).

The clinical population, on the other hand, was only used for performance evaluation. It included 20 subjects of which 2 were healthy, 2 hypertrophic, 11 infarcted

Algorithm 2: Matching Algorithm: *SPASM*

```
InitialMesh←Initialize mean Mesh;
repeat
    Intersect (ImageStack, InitialMesh );
    for all intersection points do
        Find closest mesh vertes;
    end
    CountourStack←Create 2D contours;
    Candidates (CountourStack,LearnedProfiles);
    for all possible profile positions do
        Mahalanobis (LearnedProfiles,SampledProfiles);
    end
    CandidatePoints←Smallest Mahalanobis;
    ForcePropagation (CanditatePoints);
    for all CandidatePoints do
        UpdateVectors←Calculate weight function w;
    end
    Forces←Project UpdateVectors to surface normals;
    DeformedShape←Apply forces to mesh;
    NewValidInstance (DeformedShape);
    BestFit←Best parameters to fit DeformedShape;
until iterations completed or convergence achieved ;
```

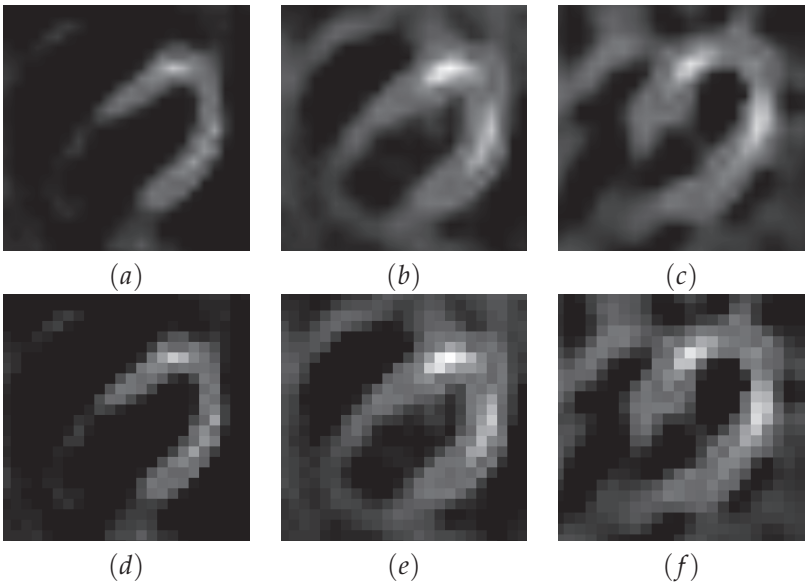


Figure 5.1: Interpolated (top) and original (bottom) axial views of a virtual (a-b, d-e) and a clinical (c,f) gSPECT study. They were reconstructed by means of OSEM (a,d) and FBP (b-c, e-f).

and 5 dilated. A rest gSPECT study and an MRI study were obtained for each subject with a mean interval of 53 days given no change in clinical condition.

Gated SPECT studies were acquired at a rate of eight frames per cardiac cycle. Patients were imaged one hour after administration of 99mTc-tetrofosmin using a Siemens ECAM SPECT system (Siemens Medical Systems, Illinois, USA) or an ADAC CardioEpic system (Philips Medical Systems, Best, NL) both with a double-detector at 90° with high resolution collimators. Sixty-four projections of a 64×64 matrix over 180° arc were obtained with a 6.60 mm/pixel resolution. Image data was reconstructed with *Filtered Back-projection* (see Figure 5.1). MRI studies were acquired using a General Electric Signa CV/i, 1.5 T scanner (General Electric, Milwaukee, USA). Datasets contained short-axis image stacks at 30 temporal phases. The slice thickness was 8 mm with an in-plane pixel resolution of 0.78mm × 0.78mm.

5.4 Methods

In the current section our methodology for automatic construction of intensity models for 3D-ASM is described thoroughly. For an overall view of the complete pipeline, refer to Figure 5.2.

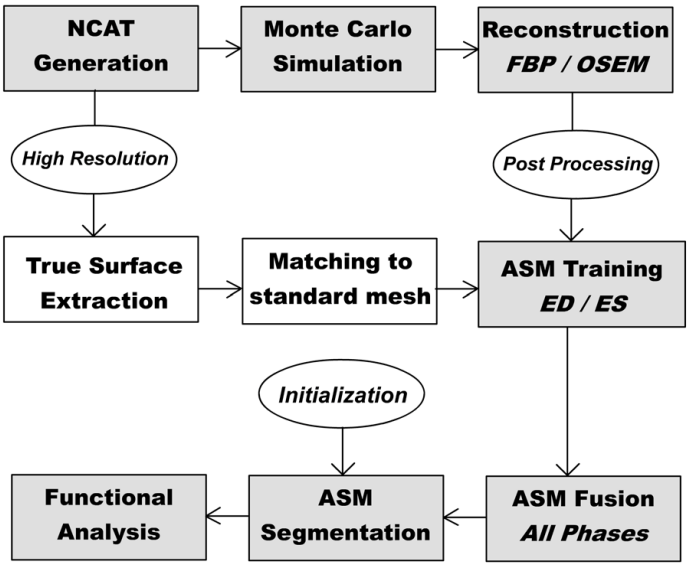


Figure 5.2: Overall description of the pipeline for construction of 3D-ASM intensity models. Main steps are represented in grey blocks and complementary steps in white ones.

5.4.1 Digital Phantoms

To ensure a realistic representation of a clinical population, several anatomical parameters were modified in a random manner, as proposed by He *et al.* [116], resembling a normal distribution obtained from the Emory PET thorax model database [117]. The minimal population size, n, was calculated following the criteria exposed by Jain *et al.* [118]. In our case, close to twenty parameters were modified during patient generation, yielding $n_{min} = 200$. Detailed description of modified parameters follow.

Anatomical Variations

Aiming to include anatomical variations which induce usual attenuation artifacts (i.e. breasts or high diaphragms) [119], three main anatomical groups were implemented (see Figure 5.3):

- *Normal Subjects:* Featuring males with a flat diaphragm and females with small breasts.
- *Male Subjects with High Liver Dome:* Half the male subjects present a high liver dome, creating strong edges which may attract segmentation algorithms.

Table 5.1: Torso parameters of female and male subjects.

Gender	Model	Body		Ribcage	
		LA	SA	LA	SA
		cm	*cm*	*cm*	*cm*
Female	F1	29	18	22	14
	F2	31	21	23	16
	F3	33	24	24	19
	F4	32	26	26	20
Male	M1	39	22	25	17
	M2	37	26	22	18
	M3	35	27	25	21
	M4	38	28	27	19

- *Female Individuals with Large Breasts*: Breast size, position and orientation were modified in order to represent possible attenuation effects.

In order to generate the population, eight representative individuals were chosen from the Emory PET thorax model database [117]. With these eight anatomical models, four male ($M1$, $M2$, $M3$, $M4$) and four female ($F1$, $F2$, $F3$, $F4$), a total of 208 subjects were created, for which half the males present a high liver dome and half the females were attributed large breasts. Figure 5.4 presents a graph which illustrates the general distribution of the virtual population. Parameters used as NCAT input are summarized in Table 5.1.

Heart Variations

The heart of each subject was varied by modifying its length and left ventricular basal radius. Global position was altered by inducing different orientation angles and translations of the heart along posterior-anterior (P-A) and lateral (Lat) directions. Specific parameters are summarized in Table 5.2.

Organ Uptake Ratios

Tracer uptakes of organs differ from patient to patient. To mimic this physiological condition, heart, liver, lung, kidney, spleen and background isotope uptake ratios were also modified in a random manner resembling a normal distribution of a typical clinical population [116]. Parameter values are displayed in Table 5.3.

Table 5.2: Anatomical parameters for heart variation according to gender. Adapted from [116].

Gender	Measure	Size		Orientation		Translation	
		Length	Ratio	Angle ϕ	Angle ψ	Lat	P-A
		cm		$^\circ$	$^\circ$	*cm*	*cm*
Female	Mean	7.4	3.20	27	40	5.2	-5.0
	SD	0.9	0.30	9	13	1.1	2.6
	Max	10.5	4.00	54	76	8.5	0.2
	Min	5.7	2.44	8	16	3.0	-10.6
Male	Mean	8.3	3.17	21	36	5.6	-6.4
	SD	0.9	0.40	9	12	1.1	2.6
	Max	11.6	4.32	41	73	8.0	1.2
	Min	6.6	2.29	0	15	3.5	-11.6

Table 5.3: Typical distribution of tracer uptake ratios on different organs. Adapted from [116].

Measure	Intensity	Ratio		
	Heart Value	Liver/ Heart	Lung/ Heart	Background/ Heart
Mean	1419	0.44	0.14	0.11
SD	810	0.19	0.04	0.05
Max	4236	1.30	0.25	0.29
Min	490	0.16	0.05	0.02

Phantom Generation

Each voxel phantom included activity and attenuation files for 8 phases of a normal (1 second) cardiac cycle. Each set consisted of 98 slices of 64×64 pixels with a 6.25 mm isotropic voxel size. This low resolution matches the usual conditions present in our clinical studies.

Up to this point the anatomical models included a full thorax model that incorporates structures other than the heart, which are important for realistic gSPECT simulation. Aiming to extract LV true surfaces, higher resolution images with only the LV structure were generated. They consisted of 321 slices of 512×512 pixels each, with a 0.78 mm isotropic voxel size. Once true surfaces were extracted from these datasets, our 3D model was aligned to them using a similarity transformation

through Procrustes Analysis [120]. Subsequently, nodes of the true surfaces acted as exact candidates to deform our mean shape using one iteration of the ASM algorithm. This process allowed warping the atlas model to all the training shapes in order to assure: *i*) control over the distribution of clinical parameters in our training database of heart shapes based on published data, *ii*) the same number of nodes and mesh topology for all true LV surfaces, and *iii*) the inclusion of high inter-subject and inter-phase variability during the matching process since the PDM is based on a larger database of real patient data.

5.4.2 Monte Carlo Simulation

In order to generate gSPECT studies for the virtual population, Monte Carlo simulation was employed using SIMIND code [101]. Details regarding the simulation set-up are given below.

Collimator Parameters

SIMIND allows for modeling different types of collimators. A Siemens Low Energy High Resolution (LEHR) collimator was chosen since it resembles our current clinical conditions [121]. Characteristics of such a collimator include: hexagonal shape, parallel hole collimator, radius of rotation of 20 cm, hole size of 1.24 mm, septal size of 0.90 mm and thickness of 23.6 mm.

Projection Parameters

Noise free projections were obtained by simulating 10^7 photon histories per projection. Sixty-four of them were obtained over a 180° arc, from 45° left posterior oblique to 45° right anterior oblique. Each projection consisted of a 64×64 matrix with 6.25 mm/pixel resolution. Energy resolution was set to 9% *Full-Width-at-Half-Maximum* (FWHM) at 140 KeV and energy window threshold to 15% photopeak at 140 KeV.

System Characterization

Ordered-subset Expectation Maximization (OSEM) reconstruction requires FWHM parameters to be determined (see Section 5.4.3). This was accomplished by measuring point-sources at different distances from the collimator surface. The point source response was approximated to a symmetric Gaussian by means of nonlinear least squares fitting [122].

Image Generation

Simulations were run using grid computing on a cluster facility of 20 dual-processor dual-core SGI Tezrix 210/2, 3Ghz/1333 Mhz, Intel Woodcrest processors. InnerGrid v5.0 (GridSystems, Palma de Mallorca, Spain) was employed as grid middleware. Distribution was achieved in the following manner: Each subject corresponds to eight digital phantom datasets (one for each cardiac phase), totalling 1664 digital phantom (208 subjects $\times$ 8 time frames). Each dataset includes sixty-four projections, which were distributed to different nodes of the cluster such that one node will simulate only one projection of one digital phantom. The whole set of projections was then concatenated to obtain full projection volumes. This methodology allowed us to reduce computation time from 16 hours to 48 minutes per subject. For the whole database it represented trading 5 months of calculations for about 7 days.

5.4.3 Tomographic Reconstruction

Aiming to obtain datasets with different intensity features for our model training (see Section 5.2) tomographic reconstruction after simulation was performed in two approaches: *Filtered Back-projection* (FBP) and *Ordered-subset Expectation Maximization* (OSEM).

FBP Reconstruction

Reconstruction was performed with a Butterworth filter. Its cut-off frequency was visually inspected on a range from 0.30 to 0.80 pixels^{-1} with step 0.2. Selected parameters were order 4 and cut-off frequency of 0.66 pixels^{-1}.

OSEM Reconstruction

Reconstruction was carried out using 4 subsets and 20 iterations. It also applied a quadratic penalty function using the 4 nearest neighbors of each pixel within a plane, along with the pixels adjacent to it on the slices above and below, as suggested by *Fessler* [122].

5.4.4 Post Processing

Following reconstruction, images were automatically masked for truncation artifact removal. Subsequently, they were scaled to a 100 grey level window, setting negative values to zero. Finally, they were saved in DICOM format in order to be processed by our 3D-ASM algorithm as a regular patient.

Table 5.4: Parameters used for ASM Segmentation

Description	Symbol	Value
Allowed Mode Variation	β	2σ
Number of Nodes	m	2677
Profile Length	n.a.	7
Profile Sampling Interval	n.a.	3 mm
Shape Variability	n.a.	75%
Gaussian Kernel Width	σ_p	7
Maximum Iterations	n.a.	15

5.4.5 3D-ASM Segmentation

Automatic segmentation of LV cavity was performed by means of 3D-ASM (see Section 5.2). Implementation details are provided next.

ASM Parameters

A uni-ventricular model of 2677 points (1835 for endocardium and 842 for epicardium) was used. The algorithm was set to run for 15 iterations or until the change in LV volume was not substantial between iterations ($\Delta Volume < 0.01$ mL). New model instances were generated with 75% of the total shape variability. This constrain was imposed to obtain a smooth fit to match the sparse data obtained from SPECT imaging, as apposed to CT imaging which allows for finer details. Other ASM parameters are summarized in Table 5.4.

Dynamic Studies Segmentation

Cardiac dynamics add to our segmentation process yet another challenge: Intensity profile variation per cardiac phase. The most intuitive scheme to approach this matter would be to obtain a model trained for each cardiac frame.

An alternative strategy is to perform ASM fusion [52], which has proven to be an effective technique for intensity model generation [123]. Under this methodology, only End Diastolic (ED) and End Systolic (ES) models were generated, since they represent the two most extreme circumstances on cardiac dynamics. Missing phases were obtained through a weighted fusion of ED and ES models. Weights used for each cardiac phase were set by the current heart phase index (LV contraction percentage) as logged by NCAT [124].

Model Initialization

We followed a very simple mechanism to roughly scale and position the mean shape of the model. The operator defines two epicardial points at the basal level and a third one at the apex. Corresponding anatomical landmarks of the mean shape were previously tagged by an experienced investigator. Consequently, the mean shape is aligned to the landmarks through a similarity transform. The manual interaction required for this procedure lasts about 30 seconds. In complex cases (i.e. large perfusion defects) longer interaction may be required, up to 1.5 minutes, for a correct depiction of basal and apical planes.

For the virtual population, initialization points were extracted automatically from the true shapes, thus eliminating initialization bias for a better analysis of segmentation accuracy. The clinical database, instead, was initialized by an experienced investigator, hereafter referred to as *Obs1*.

Functional Analysis

Once the shape model is correctly matched to specific image data, LV volumes both in End Diastole (EDV) and End Systole (ESV) can be calculated. Ejection Fraction (EF) can be derived from these measurements in order to evaluate systolic function of a patient.

5.5 Experimental Evaluation

5.5.1 Segmentation Accuracy

- *Idealized vs. Simulated Boundary Model:* To evaluate the advantage of using advanced simulations during training, a comparison with two idealized boundary models was performed. The first model consisted of a *step* function (ST), ranging from zero to one corresponding to a normalized intensity profile. The second model located the boundary at the maximum *gradient* (GR) of a sampled profile, as initially proposed by Cootes *et al.* [6]. Both virtual and clinical populations were segmented with these models.

- *True vs. Fitted Geometry:* Unsigned point-to-surface (P2S) errors were computed between the fitted meshes obtained with *idealized* and *simulated* boundary models and the gold standard LV surfaces. $Mean \pm SD$ values of all subjects in all temporal phases were computed.

- *Trained-tested Analysis:* To examine the influence of using the same reconstruction method both in training and segmentation stages, we performed an experiment combining *trained-tested* models. That is, a model *trained* with FBP reconstructed datasets was *tested* on an OSEM reconstructed dataset during

segmentation, and vice versa. A *Mann-Whitney U-test* [125], with a 95% confidence interval, was carried out to determine statistical significance of the differences.

- *Clinical Dataset:* Location of LV borders in SPECT datasets is quite subjective due to the blurred nature of these images (see Figure 5.1). However, to generate a proper gold standard for accuracy evaluation, LV contours were manually drawn according to a standard criterion: LV borders should be located at 40% of the maximum myocardial intensity. This value was obtained based on reported studies [126] and our clinical experience. In case of extensive perfusion defects, the human observer could modify the threshold down to 20%. Endocardial and epicardial border delineation of the LV, at ED, was performed by two observers (*Obs1, Obs2*) in two individual sessions (*S1, S2*). The resulting traces were used to: *i)* evaluate intra and inter-observer variability, and *ii)* obtain P2S errors of automatically segmented surfaces.

5.5.2 Sensitivity to Initialization

To evaluate the influence of initialization on our segmentation approach, the fitting process was performed 10 times for each virtual subject. Each set of initialization points was generated by adding a random error to the *true landmarks* of up to 6.25 mm (voxel size) along the X, Y and Z axis. P2S errors between true LV surfaces and the 10 fitted meshes with initialization error were computed. Also, volume errors were measured as the absolute difference between true volumes and calculated volumes.

5.5.3 LV Function Calculations

- *True vs. Measured Volume:* For the virtual population, volume error was measured with respect to true LV volumes at ED and ES. For the clinical population, gold standard volumes were obtained from manually traced LV contours on the paired MRI datasets.

  Agreement of measurements with gold standard values was assessed by means of Bland-Altman (B&A) plots [127]. Accuracy error was calculated as the percentage of absolute volume difference (*diff(True,Measured)*) relative to true volume.

- *Clinical Tool:* For the clinical dataset, a comparison with the most widespread clinical analysis tool, Quantitative Gated SPECT (QGS), was made. Results were analyzed taking into account previously published studies which describe QGS performance (see Table 5.8).

- *Population Subgroups:* In order to analyze the effect of perfusion defects on 3D-ASM volume calculations, we separated our clinical population into three subgroups. Categorization was performed by an expert clinician, *Obs2*, according to severity of the perfusion defect: *i)* none, *ii)* mild to moderate, and *iii)* severe.

5.6 Results

5.6.1 Quantitative

Segmentation Accuracy

Figure 5.5 shows LV edges obtained with 3D-ASM for the ST, GR, FBP and OSEM boundary models. Figure 5.6 displays two clinical cases with severe perfusion defects. LV edges obtained with 3D-ASM for all boundary models are displayed as well. Corresponding true surfaces are included on both figures.

Table 5.5 shows the results for the *Trained-tested Analysis* and the *Idealized vs. Simulated Boundary Model* analysis. The P2S errors of the segmentations performed with the *idealized* models are noticeably larger than the ones of the *simulated* boundary models. Endocardial errors were 28% larger than those of the FBP model and 20% larger than those of the OSEM model. Epicardial errors were 89% larger than those of the FBP model and 66% larger than those of the the OSEM model.

Sub-voxel accuracy was obtained with our segmentation method for both reconstruction techniques (See Table 5.5). For FBP reconstructed datasets, epicardial borders were segmented 35% more accurately than endocardial ones, while in OSEM reconstructed datasets the difference was 38%.

Figure 5.7 displays the statistical significance evaluation of the *Trained-tested Analysis*. All compared groups generated significantly different P2S errors, except for endocardial errors of *FBP-FBP* vs. *OSEM-FBP* and *ST-FBP* vs. *GR-FBP*, and epicardial errors of *ST-OSEM* vs. *GR-OSEM*.

Figure 5.8 displays P2S errors for each cardiac phase, with ED being $t = 1$ and ES being $t = 5$. Endocardial errors obtained at ED were 21% larger with respect to ES for both FBP and OSEM reconstructed datasets. On the other hand, epicardial errors were 18% smaller at ED for FBP reconstructed datasets and only 3% lower for OSEM reconstructed datasets.

Figure 5.9 shows P2S errors for each of the 17 LV AHA's segments [82]. For the FBP reconstructed datasets, errors corresponding to the basal plane were 43% larger than those of the medial plane and 56% larger than those of the apical plane. For the OSEM reconstructed datasets, the same comparison generated a 39% and 52% difference, respectively.

For the clinical population, intra and inter-observer variabilities are summarized in Table 5.6. P2S errors between 3D-ASM fitted shapes and manual delineations are

Table 5.5: Point-to-surface errors for the virtual population

Trained		**Tested**	**ENDO**		**EPI**	
			Mean	SD	Mean	SD
			mm	*mm*	*mm*	*mm*
IDEALIZED	ST	FBP	4.57	0.24	4.33	0.23
		OSEM	4.35	0.25	3.67	0.24
	GR	FBP	4.57	0.22	4.49	0.20
		OSEM	4.27	0.24	3.67	0.22
SIMULATED	FBP	FBP	3.56	0.27	2.33	0.22
		OSEM	3.70	0.29	3.00	0.29
	OSEM	FBP	3.61	0.26	2.69	0.14
		OSEM	3.57	0.27	2.21	0.17

Table 5.6: Point-to-surface errors for the clinical population

Variability		**ENDO**		**EPI**	
		Mean	SD	Mean	SD
		mm	*mm*	*mm*	*mm*
MANUAL	INTRAOBS	4.52	0.96	3.15	0.57
	INTEROBS	4.70	1.01	3.45	0.77
3D-ASM	FBP	4.69	0.78	4.15	0.75
	ST	5.11	0.93	6.16	1.52
	GR	5.26	0.98	4.88	1.20

Table 5.7: Sensitivity to Initialization

Dataset	Measure	P2S		LV Function		
		ENDO	EPI	EDV	ESV	EF
		mm	*mm*	*mL*	*mL*	%
FBP	Mean	3.73	2.54	3.65	3.29	4.79
	SD	0.28	0.25	1.23	1.12	1.16
	Max	21.1	21.9	73.9	40.7	24.6
	Min	0.25	0.66	0.01	0.00	0.01
OSEM	Mean	3.74	2.40	3.85	3.17	4.70
	SD	0.28	0.20	1.22	1.18	1.2
	Max	17.90	9.43	52.9	45.8	26.9
	Min	0.29	0.31	0.01	0.00	0.00

also displayed. For endocardial errors, intra- and inter-observer variabilities were not significantly different than those obtained automatically with the FBP and ST boundary models. The GR boundary model, instead, generated significantly higher P2S errors than intra-observer variability. They were also significantly higher than those of the FBP boundary model. Epicardial errors, on the other hand, were found to be significantly different for all schemes.

Sensitivity to Initialization

Table 5.7 shows the results regarding initialization sensitivity for FBP and OSEM reconstructed datasets. For both of them, the added inaccuracy caused by initialization error was 5% for endocardial borders and 8% for epicardial ones. Volume calculations presented an average error of 3.5 mL affecting the EF measurements in 4.7%. However, maximum errors came to be as large as 22 mm for accuracy measurements and 74 mL for volume calculations.

LV Function Analysis

Figure 5.10 displays B&A plots of volume calculations for the virtual population. FBP reconstructed datasets produced EDV measurements with a 94.4% accuracy, ESV measurements with a 90.0% accuracy and EF measurements with a 90.8% accuracy. For the OSEM reconstructed datasets, accuracy calculations were: 94.5% for EDV, 90.2% for ESV, and 90.9% for EF. A further analysis of EF error relative to EDV is presented in Figure 5.11.

For the clinical population, B&A plots are displayed in Figure 5.12. 3D-ASM obtained accuracy levels of 89.5% for EDV, 87.0% for ESV, and 88.1% for EF. QGS

measurements obtained accuracy levels of 81.7% for EDV, 83.5% for ESV, and 83.9% for EF. In concrete, the B&A plots for EF calculated with 3D-ASM displayed no bias and smaller variance than those of QGS.

Figure 5.13 displays accuracy errors for the clinical population subgroups. Errors showed no obvious correlation to severity of perfusion defect. Only ESV of the *none* subgroup shows a high inaccuracy for both post processing algorithms. It must be noted that half the patients in this group (n_{total}=4) presented hypertrophic LVs with collapsing walls at ES, hence the larger errors in ESV calculations.

5.6.2 Critical Analysis

Segmentation Accuracy

Idealized models demonstrated not to be robust enough for the segmentation task evaluated during this work. Figure 5.14 illustrates this fact by displaying a bar plot of the *gradient* profile averaged over all landmarks and all datasets of each population (i.e. $n_{virtual} = 208$ and $n_{clinical} = 20$). Position *zero* in the horizontal axis indicates the location of the boundary. Due to the absence of OSEM clinical datasets, only the FBP datasets are presented. Comparisons were performed against the corresponding *gold standard* which is represented with dark bars. Light bars represent the profile with respect to the best-fit boundary position according to the FBP, GR and ST *boundary models*. It is interesting to observe that in all cases (virtual and clinical datasets) the actual best-fit profiles are more alike to the simulated profiles than to the idealized profiles. This is achieved in spite of the limitations of a simulated training set, which may not capture all the details of an actual clinical database. Similarly, the standard deviation of the difference between the gold standard and the simulated boundary models were smaller than those of the two idealized boundary models. In practical terms, it reduced P2S segmentation errors by at least 20% for endocardial borders and 66% for epicardial borders.

The *Trained-tested Analysis* showed that more accurate segmentation results are obtained when the same reconstruction method is used both in training and segmentation stages. Despite the fact that OSEM reconstruction allows for better definition of LV structures, endocardial borders are located with errors of the same magnitude as those obtained with FBP. We suspect that a substantial increase in image resolution is necessary before the apparent visual improvement of OSEM reconstructed datasets has a real impact on global quantitative parameters.

Overall decreased accuracy found on endocardial border segmentation is reasonable as the relative image resolution is lower for the inner surface of the LV. That is, the correct position of a large contour (epicardium) can be found more precisely than the position of a smaller contour (endocardium), given the same pixel size.

Greater P2S errors found at basal level are quite understandable since a correct

depiction of LV basal plane is a well known complication of cardiac imaging post-processing for most modalities [128]. SPECT images are specially challenging on this matter owing to the lack of commonly used anatomical landmarks such as the mitral valve or the left atria.

As can be observed in Figure 5.9, P2S errors are larger at the inferoseptal basal segment. Because of the presence of the membranous septum, this region displays almost no tracer activity. Hence, during fitting the mesh is not actively deformed at this area the LV wall. This is represented in the virtual phantoms as thinner septal structures. It is particularly noticeable at ED where the difference in activity between the basal portion of the lateral wall and the basal portion of the septal wall is quite visible. At ES, though, due to thickening and shortening of the LV walls, the septum can be better defined at basal levels.

For the cardiac phase analysis, the larger epicardial P2S errors found at ES phase are natural (lower resolution and partial volume effect). However, the decrease in error observed for endocardial borders is counterintuitive. Visual inspection suggests this is caused by the higher segmentation inaccuracy at basal level, as mentioned above.

For the clinical studies, 3D-ASM errors for endocardial borders are comparable to inter-observer variability. However, epicardial boundaries presented 20% larger errors than inter-observer variability. This might be due to overestimation of wall thickness in places of extensive perfusion defects. Regardless of lack of data, a human observer may deduct a thinning of the LV walls caused by chronic infarcted myocardium. ASM, on the other hand, will try to conserve the wall thickness present on the remaining sampled data. It must noted that intra and inter-observer variability under *uncontrolled* circumstances (i.e. without a standardized criterion) will most likely be larger than the ones measured during our experiments.

Sensitivity to Initialization

The evaluation of initialization sensitivity illustrated the extent of inaccuracy caused by initialization error. Yet, in average, this inaccuracy was rather small. The maximum errors revealed noticeable bias in case of very improper initialization points. However, in clinical dataset processing, initialization would be performed by a trained technician capable of efficiently and correctly defining basal and apical positions.

LV Function Analysis

For the virtual population, the scatter distribution of the B&A plots showed a dependency of the error on the LV volume. B&A plots also revealed that our algorithm tends to underestimate EDV, a tendency also present on QGS (See Table 5.8). The most extreme case of overestimation was found for the largest heart. Yet its

Table 5.8: Meta Analysis of published works comparing QGS postprocessing results against a gold standard.

Author	Year	N	Population	Gold Standard (GS)	Post Processing Software						Small Hearts	
					QGS			GS			QGS	
					EDV	ESV	EF	EDV	ESV	EF	EDV	EF
		Total (F/M)			*m*L	*m*L	%	*m*L	*m*L	%	⇑ / ⇓	
Achtert *et al.* [112]	1998	3 (*n*.a.)	Normal	MCAT	111.0	42.5	61.9	121.8	50.0	59	⇓	⇑
Tadamura *et al.* [129]	1999	16 (3/13)	Surgery	MRI	106.1	54.9	47.9	112.1	55.3	50.8	*n*.a.	*n*.a.
Bavelaar-Croon *et al.* [130]	2000	21 (7,14)	CAD	MRI	151.0	97.0	43.0	191.0	114.0	45.0	*n*.a.	*n*.a.
Nakajima *et al.* [131]	2001	4 (*n*.a.)	Normal	Phantom	123.0	*n*.a.	*n*.a.	131.0	*n*.a.	*n*.a.	⇓	⇑
Nakajima *et al.* [131]	2001	30 (10/20)	Mixed	GBP	98.0	*n*.a.	54.0	103.0	*n*.a.	49.0	⇓	⇑
Lipke *et al.* [132]	2004	54 (15/39)	CAD	MRI	122.0	62.0	52.2	139.0	60.0	60.0	*n*.a.	*n*.a.
Lomsky *et al.* [133]	2005	5 (*n*.a.)	Normal	NCAT	113.0	64.0	45.6	115.2	44.8	61.0	⇓	⇑
Schaefer *et al.* [134]	2005	70 (16/54)	CAD	MRI	120.0	60.0	53.2	137.0	57.0	60.6	*n*.a.	*n*.a.
Stegger *et al.* [111]	2007	70 (16/54)	CAD	MRI	120.0	60.0	53.0	137.0	57.0	61.0	⇓	⇑
Wu *et al.* [135]	2007	33 (*n*.a.)	CAD	MRI	*n*.a.	*n*.a.	40.2	*n*.a.	*n*.a.	40.1	*n*.a.	*n*.a.

n.a.= Not available; Normal= Healthy hearts; Surgery= Patients who underwent coronary artery bypass surgery; CAD= Known or suspected coronary arterial disease; Mixed= Common cardiomyopathies; MCAT= 3D dynamic cardiac-torso phantom; MRI= Magnetic resonance imaging; Phantom= Cylindrical mathematical model; GBP= Gated blood-pool study; NCAT= 4D NURBS-based cardiac-torso phantom; ⇓= Underestimation; ⇑= Overestimation.

difference is within reported limits of discrepancy (30 mL from gold standard measurements) [136].

For ESV, a slight overestimation is revealed through the B&A plots, previously stated for QGS as well (Table 5.8). For EF, the confidence intervals in the B&A plots are wider than those for EDV and ESV, probably caused by the higher dispersion observed on lower EF values. Note in Figure 5.11 that many of the large discrepancies in EF calculations are located around small hearts (50 mL EDV). This parameter is known to be overestimated for this type of hearts when calculated from perfusion studies [137]. This is attributed to artificially increased counts in the LV cavity, complicating a proper calculation of ESV volumes.

For the clinical population, overall patterns of B&A plots were comparable to those of QGS. Calculated parameters showed less biased underestimations. Smaller confidence intervals were found for 3D-ASM for all calculated parameters. Similarly, accuracy levels were higher than those obtained with QGS for all measured parameters.

No obvious correlation between perfusion defect severity and segmentation inaccuracy was found for our clinical database. Inaccuracy could be more related to low image quality or segmentation difficulty depending on pathology. For instance, the group with no perfusion defects was composed of hypertrophic patients and one dilated patient with Left Bundle Branch Block, both difficult cases to segment even for a human observer.

5.7 Discussion

5.7.1 Clinical Contributions

Our method obtained higher accuracy compared to QGS, one of the most widespread commercial packages. Although this result is obtained in a small population, this is quite encouraging for a *simulation based* approach since it bypasses the labor of clinical database collection and, furthermore, the underlying methodology is potentially applicable to other modalities.

The employed segmentation method could either be applied on transaxial slices or on reformatted short axis images. The use of the transaxial slices is preferable since time consuming operator assistance is required to define the LV long axis.

As can be concluded from previous works (see Table 5.8) the tendencies of QGS for small hearts still needs further review. Virtual populations with specific heart sizes may be useful for investigating this matter.

5.7.2 Outlook

The feasibility of our approach has been illustrated in the context of one clinical application (viz. cardiac image analysis) and one specific imaging modality (viz.

gSPECT). Nevertheless, the potential of this approach is much broader.

To start off, it can help decoupling the sample size requirements of building relevant statistics for the intensity models. Shape models could be built based on a high-resolution imaging modality (e.g. CT) and the derived PDM be sampled to generate a virtual population from which simulated images of other modalities can be produced (e.g. MR, SPECT or US). Regarding sample size, only few real clinical images might be available for extreme anatomical variants (e.g. very small or very large hearts). However, they can be sampled uniformly when creating the virtual population for simulated data.

Another problem in learning intensity models directly from real images is related to the rapid evolution of most imaging technologies. Handling this problem would become simpler with our technique as we can regenerate the intensity models, as long as the employed simulator allows for it. The upgrades can be related to: *i)* improvement of spatial resolution (i.e. smaller pixel size), *ii)* increase of temporal resolution (i.e. more frames per cycle), *iii)* development of better reconstruction techniques (e.g. iterative algorithms), *iv)* isotropic voxels (i.e. for MRI or CT), *v)* variation on physical parameters used during acquisition (e.g. modification of MRI sequences), etc.

As the final advantage, we would like to mention that avoiding the need to use shapes derived from manually contoured shapes prevents expert dependency as the true boundary information is known by construction. Moreover, the possibility to build intensity models in every major modality based on a high-resolution PDM pave the way for handling more consistently multimodal datasets.

This approach, however, may present a number of disadvantages, depending on the realism and accuracy of the image acquisition simulator, such as: computationally expensive processing, large amount of input parameters sometimes hard to determine, use of theoretical noise which may not resemble clinical conditions, etc.

5.8 Conclusion

This paper introduced the notion of using advanced imaging simulators to enable automatic creation of intensity models. Results show that gSPECT studies can be successfully segmented by models trained under this scheme with sub-voxel accuracy. The accuracy in estimated LV function parameters range from 90.0% to 94.5% for the virtual population and from 87.0% to 89.5% for the clinical population. These results are within the intervals reported by other widespread clinical segmentation tools.

Our future efforts along the generic approach we presented here is to extend this technique to other imaging modalities. Efforts are underway to apply this approach to 3D US data [138] and we do not foresee fundamental issues not to extend this technique to MRI and CT.

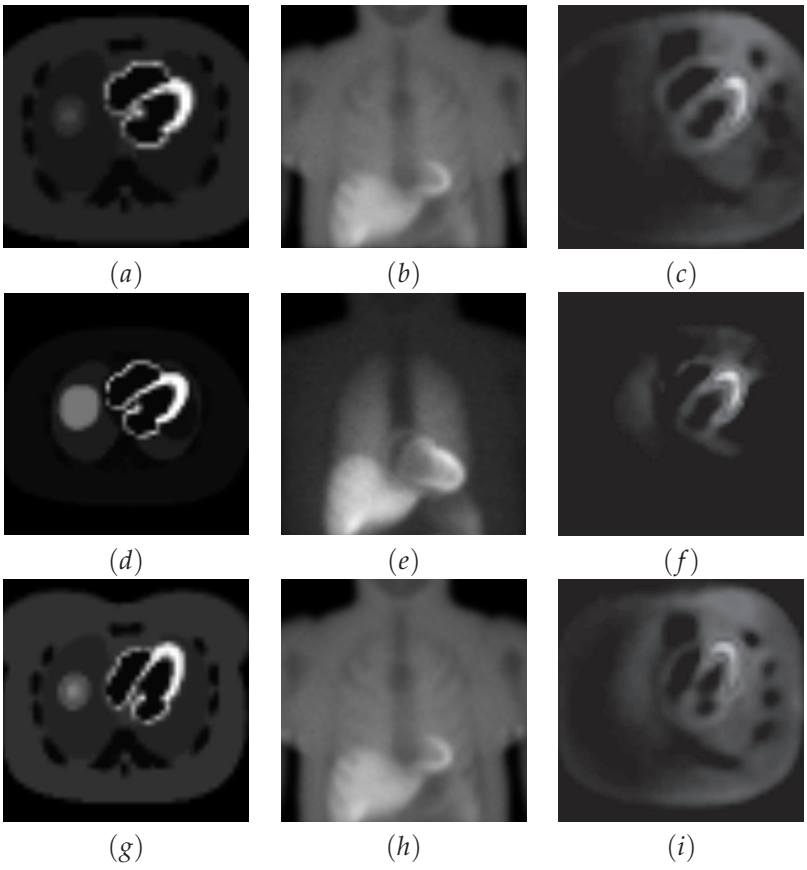


Figure 5.3: Sample of the three anatomical groups: Normal subjects (a-c), male subjects with high liver dome (d-f) and female individuals with large breasts (g-i). Images were generated with NCAT (left), SIMIND (middle) and ASPIRE (right), respectively.

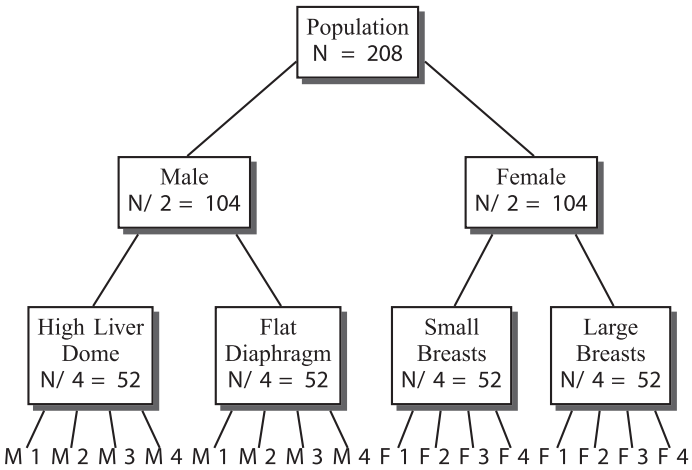
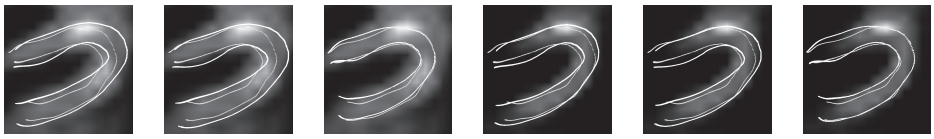
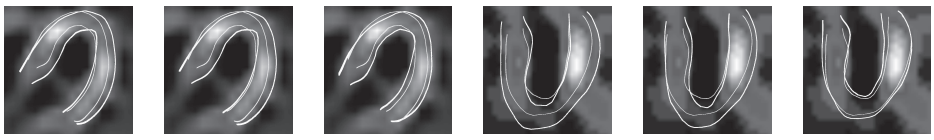


Figure 5.4: General distribution of the virtual population, subdivided into anatomical groups. See Section 5.4.1 for details.



(*a*) (*b*) (*c*) (*d*) (*e*) (*f*)

Figure 5.5: Axial view of a virtual study for FBP (a-c) and OSEM (d-f) reconstructed images. Edges obtained automatically by 3D-ASM with ST (a,d), GR (b,e), FBP (c) and OSEM (f) boundary models are shown in white (thick). True edges are displayed on yellow (thin).



(*a*) (*b*) (*c*) (*d*) (*e*) (*f*)

Figure 5.6: Two clinical cases with severe perfusion defects: Case one in axial view (a-c) and case two in long-axis view (d-f). Edges obtained automatically by 3D-ASM with ST (a,d), GR (b,e) and FBP (c,f) boundary models are shown in white (thick). True edges are displayed on yellow (thin).

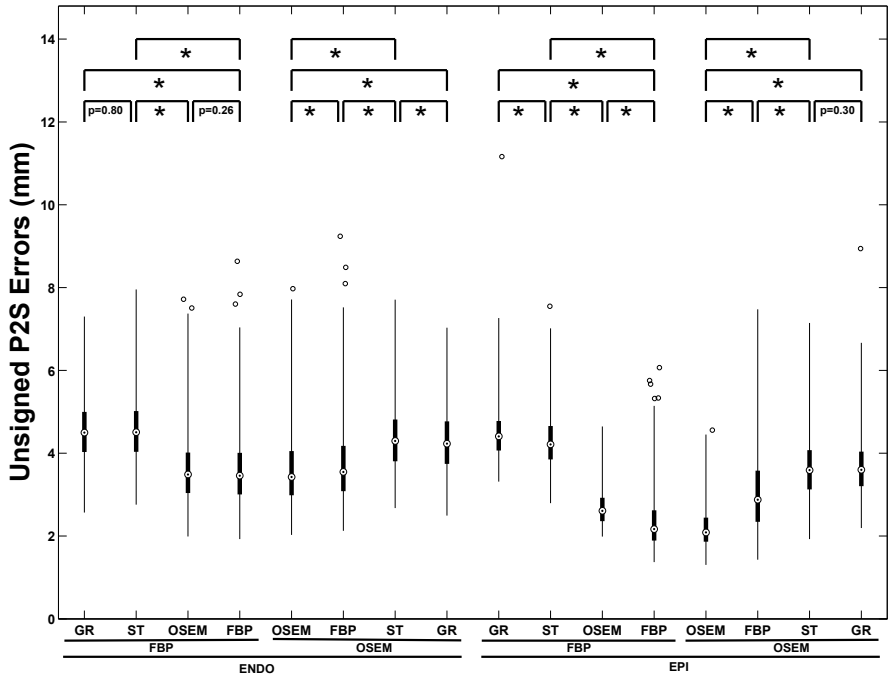


Figure 5.7: Box-and-whisker plot of the *Trained-tested Analysis* for FBP, OSEM, ST and GR boundary models. Connecting lines illustrate compared groups. The stars represent statistically significant differences. p values of the statistically nonsignificant differences are also displayed.

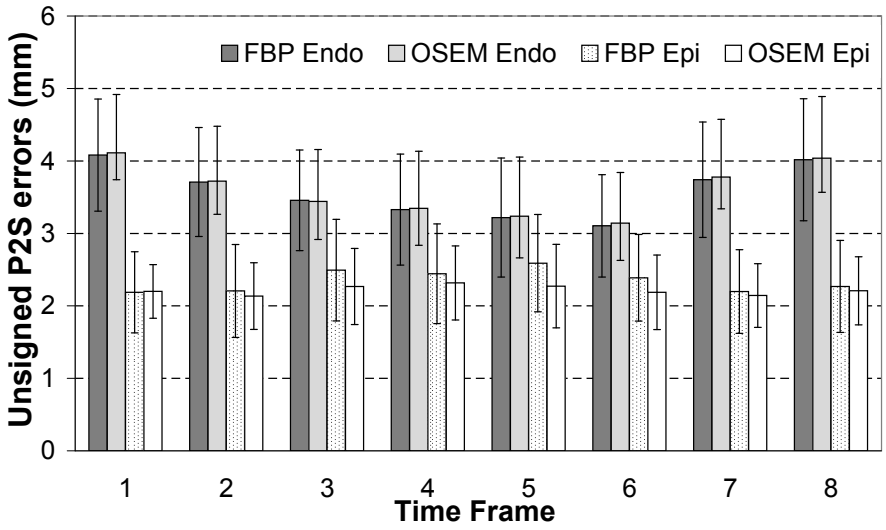


Figure 5.8: Bar plot of mean point-to-surface errors per cardiac phase for FBP and OSEM reconstructed datasets. ED corresponds to $t = 1$ and ES to $t = 5$. Error bars represent SD of the measurements.

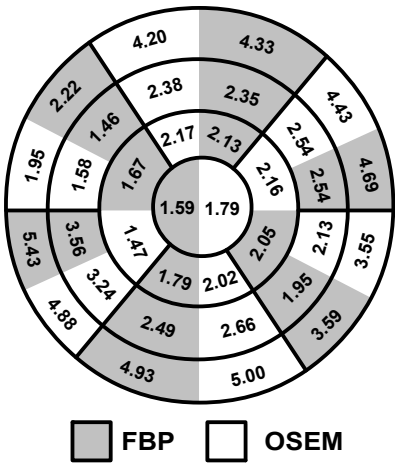


Figure 5.9: Bull's eye plot of point-to-surface errors for each of the 17 Left Ventricular AHA's segments for FBP and OSEM reconstructed datasets.

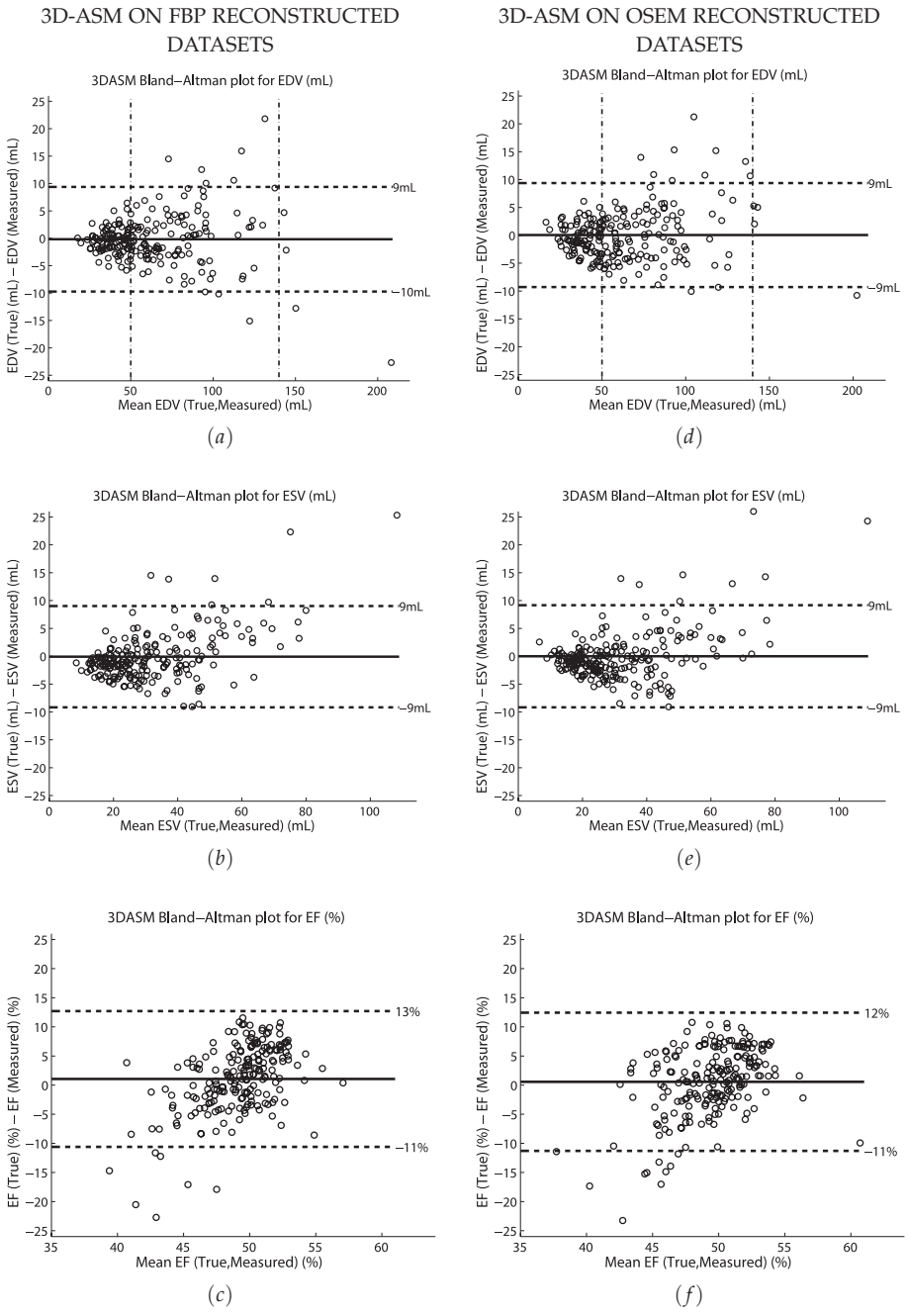


Figure 5.10: Virtual population: Bland-Altman plots for EDV (a,d), ESV (b,e) and EF (c,f) comparing gold standard and measured values estimated with 3D-ASM for the datasets reconstructed by means of FBP (top) and OSEM (bottom).

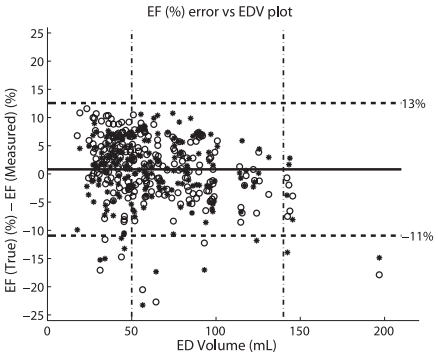


Figure 5.11: Plot of Ejection Fraction (EF) error vs End Diastolic (ED) volume for FBP (o) and OSEM (*) reconstructed datasets of the virtual population.

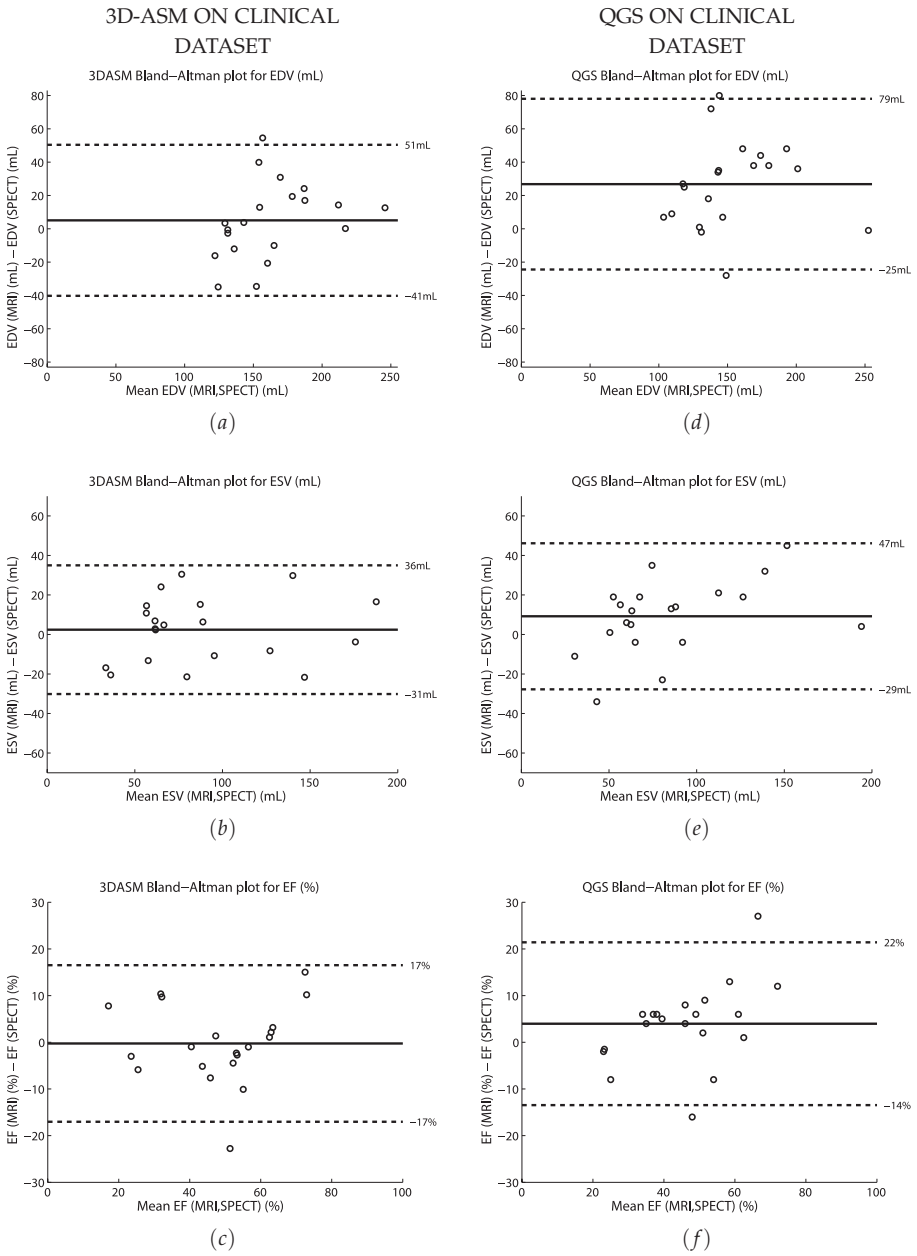


Figure 5.12: Clinical population: Bland-Altman plots for EDV (a,d), ESV (b,e) and EF (c,f) comparing gold standard and measured values estimated with 3D-ASM (top) and QGS (bottom).

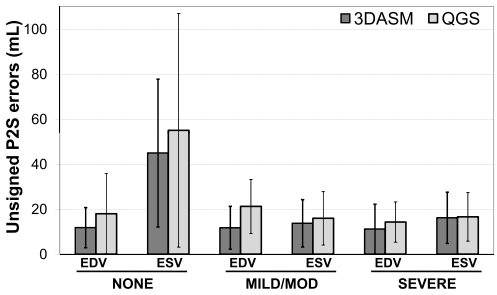


Figure 5.13: Accuracy errors on volume calculations for the three population subgroups according to perfusion defect severity. EDV and ESV errors for 3D-ASM and QGS. Error bars represent SD.

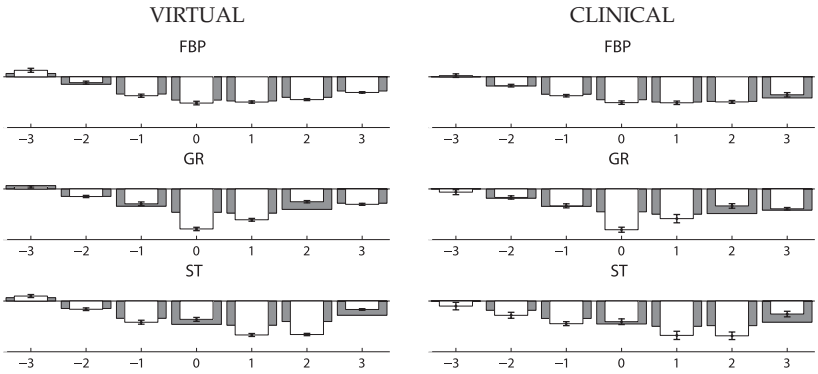


Figure 5.14: Bar plot comparing the underlying *gold standard* and the best-fit profiles using the three boundary models in both virtual (top) and clinical (bottom) populations. Plots show the gradient profiles with respect to the gold standard boundary position (zero abscissa). Dark and light bars stand for the mean gold standard gradient profile (around and along the normal to expert surfaces) and the best-fit gradient profiles (around and along the normal to candidate surfaces based on FBP, GR and ST boundary models), respectively. Error needles on the light bars represent the SD of the difference between the gold standard and model gradient profiles. Means and SDs were computed over all landmarks and all datasets for both populations. Experiments show that the higher accuracy achieved with our proposed technique is consistent with a more accurate modeling of gradient profiles.

APPENDIX A

Unbiased covariance matrix estimate in the general case

The unbiased estimate of the covariance matrix of a set of N independent random observations $\mathbf{z}_i$ $(i = 1, \ldots, N)$ is

$$\mathbf{S} = \frac{1}{1 - \sum_{i=1}^{N} p_i^2} \cdot \sum_{i=1}^{N} p_i \left(\mathbf{z}_i - \bar{\mathbf{z}}\right) \left(\mathbf{z}_i - \bar{\mathbf{z}}\right)^T \tag{A.1}$$

where $\bar{\mathbf{z}} = \sum_{i=1}^{N} p_i \mathbf{z}_i$ and p_i is the probability associated with the i-th observation $\mathbf{z}_i$. This result is easily obtained following the same steps as in [27], without assuming all the probabilities being equal to $\frac{1}{N}$. Note that when $p_i = \frac{1}{N}$, the estimate (A.1) reduces to the usual unbiased estimate used in statistics and derived in [27].

APPENDIX B

Linearity of the Warp

Lemma 1. *Given an arbitrary set of linear transformations $L_i : \mathbb{R}^n \to \mathbb{R}, i = 1, \ldots, K$ of a n-dimensional vector $\mathbf{x}$ into the set of real numbers, the transformation $\varphi : \mathbb{R}^n \to \mathbb{R}^K$, such that $\varphi(\mathbf{x}) = [L_1(\mathbf{x}), L_2(\mathbf{x}), \ldots, L_K(\mathbf{x})]^T$, is a linear transformation.*

Proof. Let $\alpha \in \mathbb{R}$. Then

$$\begin{aligned}\varphi(\alpha\mathbf{x}) &= [L_1(\alpha\mathbf{x}), L_2(\alpha\mathbf{x}), \ldots, L_K(\alpha\mathbf{x})]^T = \\ &= \alpha \cdot [L_1(\mathbf{x}), L_2(\mathbf{x}), \ldots, L_K(\mathbf{x})]^T = \alpha \cdot \varphi(\mathbf{x})\end{aligned}$$

Let $\mathbf{y} \in \mathbb{R}^n$. Then

$$\begin{aligned}\varphi(\mathbf{x}+\mathbf{y}) &= [L_1(\mathbf{x}+\mathbf{y}), L_2(\mathbf{x}+\mathbf{y}), \ldots, L_K(\mathbf{x}+\mathbf{y})]^T = \\ &= \varphi(\mathbf{x}) + \varphi(\mathbf{y})\end{aligned}$$

The lemma is proven. □

The transformation φ can be, without loss of generality, considered a texture warp with an interpolation, that is a linear function of pixels, and $\mathbf{x}$ a texture vector. Then L_i will be the pixels obtained by interpolating between some of the elements of $\mathbf{x}$.

APPENDIX C

Orthonormality of eigenvectors

Our goal is to demonstrate that there is no need to recompute the eigendecomposition of the matrix $\tau_i\left(\mathbf{\Phi}_{gi}\right)\mathbf{\Lambda}_{gi}\left[\tau_i\left(\mathbf{\Phi}_{gi}\right)\right]^T$ in order to fuse the eigenspaces $\widetilde{\Omega}_{gi} = \left(\tau_i\left(\bar{\mathbf{g}}_i\right), \tau_i\left(\mathbf{\Phi}_{gi}\right), \mathbf{\Lambda}_{gi}, N_i\right), i = 1, \ldots, M$. In other words it is irrelevant for the fusion whether $\tau_i\left(\mathbf{\Phi}_{gi}\right)$ is orthonormal or not.

Let us start with mentioning that since $\tau_i\left(\mathbf{g}_{ij}\right)$ is linear with respect to $\mathbf{g}_{ij}$ then there exists a matrix $\mathbf{A}_i$ such that [28]:

$$\tau_i\left(\mathbf{g}_{ij}\right) = \mathbf{A}_i\mathbf{g}_{ij}$$

Let the covariance matrix of $\mathbf{g}_{ij}$, for a given i, be

$$\mathbf{G}_i = \mathbf{\Phi}_{gi}\mathbf{\Lambda}_{gi}\mathbf{\Phi}_{gi}^T$$

Now let us obtain a covariance matrix for the warped observations (2.22).

$$\begin{aligned}\widetilde{\mathbf{G}}_i &= \frac{1}{N_i - 1}\sum_{j=1}^{N_i}\left[\tau_i\left(\mathbf{g}_{ij}\right) - \tau_i\left(\bar{\mathbf{g}}_i\right)\right]\left[\tau_i\left(\mathbf{g}_{ij}\right) - \tau_i\left(\bar{\mathbf{g}}_i\right)\right]^T = \\ &= \frac{1}{N_i - 1}\sum_{j=1}^{N_i}\mathbf{A}_i\left[\mathbf{g}_{ij} - \bar{\mathbf{g}}_i\right]\left[\mathbf{g}_{ij} - \bar{\mathbf{g}}_i\right]^T\mathbf{A}_i^T = \\ &= \mathbf{A}_i\mathbf{G}_i\mathbf{A}_i^T = \mathbf{A}_i\mathbf{\Phi}_{gi}\mathbf{\Lambda}_{gi}\left(\mathbf{A}_i\mathbf{\Phi}_{gi}\right)^T \end{aligned} \tag{C.1}$$

By eigendecomposition

$$\widetilde{\mathbf{G}}_i \stackrel{\Delta}{=} \widetilde{\mathbf{\Phi}}_{gi}\widetilde{\mathbf{\Lambda}}_{gi}\widetilde{\mathbf{\Phi}}_{gi}^T$$

Let us demonstrate that

Lemma 2. *For any matrix* $\mathbf{A}$ *and any nonsingular diagonal matrix* $\mathbf{\Lambda}$

$$r\left[\mathbf{A}\mathbf{\Lambda}\mathbf{A}^T\right] = r\left(\mathbf{A}\right)$$

Proof. It is known that $r\left[\mathbf{B}\mathbf{B}^T\right] = r\left[\mathbf{B}\right]$ for any matrix $\mathbf{B}$ [29]. It is also known [29] that multiplication of a matrix by a nonsingular matrix does not change the rank of that matrix. Let $\mathbf{B} = \mathbf{A}\mathbf{\Lambda}^{\frac{1}{2}}$. Then

$$r\left[\mathbf{A}\mathbf{\Lambda}\mathbf{A}^T\right] = r\left[\mathbf{B}\mathbf{B}^T\right] = r\left[\mathbf{B}\right] = r\left[\mathbf{A}\mathbf{\Lambda}^{\frac{1}{2}}\right] = r\left[\mathbf{A}\right]$$

The lemma is thus proven. □

Looking at (2.6) one can see that the fusion is essentially a linear combination of covariance matrices, so, taking into account the need of warping each texture onto the fused mean shape, each of these covariance matrices will be transformed (as in (C.1)), which is equivalent to transforming only the eigenvectors in their eigendecomposition. The only place where the eigenvectors are used specifically is in the construction of the matrix $\mathbf{H}$. But

1. $\mathbf{H}$ is orthonormalized;
2. $\widetilde{\mathbf{\Phi}}_{gi}$ and $\mathbf{A}_i\mathbf{\Phi}_{gi}$ span the subspace of the same dimensionality (directly follows from the above lema);
3. equation (2.6) does not require that the factorization of the covariance matrices is carried out by the eigendecomposition;
4. from (2.22) it follows that any warped observation can be represented by the warped eigenvectors. In other words, warped eigenvectors span the subspace of the warped observations.

Therefore, there is no specific need to recompute the eigendecomposition of the transformed covariance matrices.

APPENDIX D

Coordinate Transformation in Vector Spaces

Let $L : \mathbb{V} \rightarrow \mathbb{W}$ be a linear transformation of an n-dimensional vector space $\mathbb{V}$ into an m-dimensional vector space $\mathbb{W}$ ($m \neq 0$, $n \neq 0$). And let matrices $\mathbf{V} = [\mathbf{v_1}|\mathbf{v_2}|\dots|\mathbf{v_n}]$ and $\mathbf{W} = [\mathbf{w_1}|\mathbf{w_2}|\dots|\mathbf{w_m}]$ be the bases for $\mathbb{V}$ and $\mathbb{W}$ respectively. Then, there exists a $m \times n$ matrix $\mathbf{A}$ such that [28], [29]

$$[L(\mathbf{x})]_W = \mathbf{A} \cdot [\mathbf{x}]_V, \quad \mathbf{x} \in \mathbb{V}$$

where

$$A = [[L(\mathbf{v_1})]_W, [L(\mathbf{v_2})]_W, \dots, [L(\mathbf{v_n})]_W] \tag{D.1}$$

Notation $[\mathbf{x}]_V$ means that vector $\mathbf{x}$ is represented in the basis $\mathbf{V}$, the same for $[\mathbf{x}]_W$.

In other words, (D.1) means that columns of the transformation matrix are the basis vectors of $\mathbb{V}$ transformed by the linear map L and written in the basis of $\mathbb{W}$.

If $\mathbb{V} = \mathbb{R}^n$ and $\mathbb{W} = \mathbb{R}^m$, then for $\mathbf{x} \in \mathbb{V}$ and $\mathbf{y} = L(\mathbf{x}) \in \mathbb{W}$ there exist two vectors $\mathbf{a}$ and $\mathbf{b}$ such that $\mathbf{x} = \mathbf{V} \cdot \mathbf{a}$ and $\mathbf{y} = \mathbf{W} \cdot \mathbf{b}$.

What is needed is the relationship between $\mathbf{a}$ and $\mathbf{b}$, which are the coordinates of vectors $\mathbf{x}$ and $\mathbf{y}$ in the bases of $\mathbb{V}$ and $\mathbb{W}$. Therefore, recalling (D.1) it can be written:

$$\begin{aligned} \mathbf{A} &= \mathbf{W}^{-1} \cdot [L(\mathbf{v}_1), L(\mathbf{v}_2), \dots, L(\mathbf{v}_n)] \\ \mathbf{b} &= \mathbf{A} \cdot \mathbf{a} \end{aligned} \tag{D.2}$$

If L is the identity transformation then $\mathbf{y} \equiv \mathbf{x}$ and obviously $\mathbf{A} = \mathbf{W}^{-1} \cdot \mathbf{V}$.

Bibliography

[1] P. Ekman, E. R. Sorenson, and W. V. Friesen, "Pancultural elements in facial displays of emotion," *Science*, vol. 164, no. 3875, pp. 86–88, 1969.

[2] P. Ekman and W. V. Friesen, "Constants across culture in the face and emotion," *Journal of Personality and Social Psychology*, vol. 17, pp. 124–129, 1971.

[3] P. Ekman, "Universal and cultural differences in facial expression of emotion," in *Nebraska Symposium on Motivation*, J. R. Cole, Ed., vol. 19, 1972, pp. 207–283.

[4] S. Allender, P. Scarborough, V. Peto, M. Rayner, J. Leal, R. Luengo-Fernandez, and A. Gray. (2009, 18th May) European cardiovascular disease statistics 2008. [Online]. Available: http://www.heartstats.org/datapage.asp?id=7683

[5] A. A. Young and A. F. Frangi, "Computational cardiac atlases: from patient to population and back." *Exp. Physiol.*, vol. 94, no. 5, pp. 578–596, 2009.

[6] T. F. Cootes, D. Cooper, C. J. Taylor, and J. Graham, "Active shape models - their training and application," *Comput. Vis. Image Understand.*, vol. 61, no. 1, pp. 38–59, 1995.

[7] T. Cootes and C. Taylor, "Active shape models – smart snakes," in *Proc. British Machine Vision Conf.*, 1992, pp. 266–275.

[8] T. Cootes, G. Edwards, and C. J. Taylor, "Active appearance models," in *Proc. European Conf. on Computer Vision, LNCS vol. 1407*, 1998, pp. 484–498.

[9] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 681–685, 2001.

[10] S. C. Mitchell, B. P. Lelieveldt, R. J. van der Geest, H. G. Bosch, J. H. Reiber, and M. Sonka, "Multistage hybrid active appearance model matching: segmentation of left and right ventricles in cardiac mr images," *IEEE Trans. Med. Imag.*, vol. 20, no. 5, pp. 415–423, 2001.

[11] S. C. Mitchell, J. G. Bosch, B. P. Lelieveldt, R. J. van der Geest, J. H. Reiber, and M. Sonka, "3-D active appearance models: segmentation of cardiac MR and ultrasound images," *IEEE Trans. Med. Imag.*, vol. 21, no. 9, pp. 1167–1178, 2002.

[12] 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," *IEEE Trans. Med. Imag.*, vol. 21, no. 8, pp. 924–933, 2002.

[13] I. Matthews and S. Baker, "Active Appearance Models revisited," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 135–164, 2004.

[14] A. U. Batur and M. H. Hayes, "Adaptive active appearance models," *IEEE Trans. Image Process.*, vol. 14, no. 11, pp. 1707–1721, 2005.

[15] R. Beichel, H. Bischof, F. Leberl, and M. Sonka, "Robust active appearance models and their application to medical image analysis." *IEEE Trans. Med. Imag.*, vol. 24, no. 9, pp. 1151–1169, 2005.

[16] F. M. Sukno, J. J. Guerrero, and A. F. Frangi, "Homographic active shape models for view-independent facial analysis," in *Proc. SPIE Biometric Technology for Human Identification II*, A. K. Jain and N. K. Ratha, Eds., vol. 5779, 2005, pp. 152–163.

[17] R. Donner, M. Reiter, G. Langs, P. Peloschek, and H. Bischof, "Fast active appearance model search using canonical correlation analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1690–1694, 2006.

[18] T. F. Cootes, C. J. Twining, K. O. Babalola, and C. J. Taylor, "Diffeomorphic statistical shape models," *Image and Vision Computing*, vol. 26, no. 3, pp. 326–332, 2008.

[19] H. C. van Assen, M. G. Danilouchkine, M. S. Dirksen, J. H. C. Reiber, and B. P. F. Lelieveldt, "A 3-D active shape model driven by fuzzy inference: Application to cardiac CT and MR," vol. 12, no. 5, pp. 595–605, 2008.

[20] P. Hall, D. Marshall, and R. Martin, "Merging and splitting eigenspace models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 9, pp. 1042–1049, 2000.

[21] H. C. van Assen, M. G. Danilouchkine, A. F. Frangi, S. Ordas, J. J. Westenberg, J. H. Reiber, and B. P. Lelieveldt, "SPASM: a 3D-ASM for segmentation of sparse and arbitrarily oriented cardiac MRI data," *Med. Image Anal.*, vol. 10, no. 2, pp. 286–303, 2006.

[22] J. Lim, D. Ross, R.-S. Lin, and M.-H. Yan, "Incremental learning for visual tracking," in *Advances in Neural Information Processing Systems*, vol. 17, 2005.

[23] A. Levy and M. Lindenbaum, "Sequential Karhunen-Loeve basis extraction and its application to images," *IEEE Trans. Image Process.*, vol. 9, no. 8, pp. 1371–1374, 2000.

[24] J. Bunch, C. Nielsen, and D. Sorenson, "Rank-one modification of the symmetric eigenproblem," *Numer. Math.*, vol. 31, pp. 31–48, 1978.

[25] S. Chandrasekaran, B. Manjunath, Y. Wang, J. Winkler, and H. Zhang, "An eigenspace update algorithm for image analysis," *Graph. Model Im. Proc.*, vol. 59, no. 5, pp. 321–332, 1997.

[26] R. DeGroat and R. Roberts, "Efficient, numerically stablized rank-one eigenstructure updating," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 2, pp. 301–316, 1990.

[27] H. Murakami and B. Kumar, "Efficient calculation of primary images from a set of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 4, no. 5, pp. 511–515, 1982.

[28] A. Franco, A. Lumini, and D. Maio, "Eigenspace merging for model updating," in *Proc. Int. Conf. on Pattern Recognition*, vol. 2, 2002, pp. 156–159.

[29] X. S. Zhou, D. Comaniciu, and A. Gupta, "An information fusion framework for robust shape tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 1, pp. 115–129, 2005.

[30] M. B. Stegmann, B. K. Ersbøll, and R. Larsen, "FAME – A flexible appearance modeling environment," *IEEE Trans. Med. Imag.*, vol. 22, no. 10, pp. 1319–1331, 2003.

[31] A. Martinez and R. Benavente, "The AR face database," CVC, Barcelona, Tech. Rep., 1998. [Online]. Available: http://rvl1.ecn.purdue.edu/_aleix/aleix face DB.html

[32] R. Davies, "Learning shape: Optimal models for analysing natural variability," Ph.D. dissertation, Division of Imaging Science and Biomedical Engineering, University of Manchester, 2002.

[33] "Equinox face database." [Online]. Available: http://www.equinoxsensors.com/products/HID.html

[34] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre, "XM2VTSDB: The extended M2VTS database," in *Proc. Int. Conf. on Audio- and Video-Based Biometric Person Authentication*, 1999, pp. 72–77. [Online]. Available: http://xm2vtsdb.ee.surrey.ac.uk/

[35] V. Perlibakas, "Distance measures for PCA-based face recognition," *Pattern Recogn. Lett.*, vol. 25, no. 6, pp. 711–724, 2004.

[36] S. Gong, S. McKenna, and J. J. Collins, "An investigation into face pose distributions," in *Proc. IEEE Int. Conf. on Automatic Face and Gesture Recognition*, 1996, pp. 265–270.

[37] F. Shih, C. Fu, and K. Zhang, "Multi-view face identification and pose estimation using B-spline interpolation," *Inform. Sciences*, vol. 169, no. 3–4, pp. 189–204, 2005.

[38] B. Gokberk, L. Akarun, and E. Alpaydin, "Feature selection for pose invariant face recognition," in *Proc. Int. Conf. on Pattern Recognition*, vol. 4, 2002, pp. 306–309.

[39] J. Huang, X. Shao, and H. Wechsler, "Face pose discrimination using support vector machines SVM," in *Proc. Int. Conf. on Pattern Recognition*, vol. 1, 1998, pp. 154–156.

[40] S. Li, Q. Fu, L. Gu, B. Scholkopf, Y. Cheng, and H. Zhang, "Kernel machine based learning for multi-view face detection and pose estimation," in *Proc. IEEE Int. Conf. on Computer Vision*, vol. 2, 2001, pp. 674–679.

[41] Y. Li, S. Gong, J. Sherrah, and H. Liddell, "Support vector machine based multi-view face detection and recognition," *Image Vis. Comput.*, vol. 22, no. 5, pp. 413–427, 2004.

[42] K. C. Lee, J. Ho, M.-H. Yang, and D. Kriegman, "Video-based face recognition using probabilistic appearance manifolds," in *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, vol. 1, 2003, pp. 313–320.

[43] K. Okada, S. Akamatsu, and C. von der Malsburg, "Analysis and synthesis of pose variations of human faces by a linear PCMAP model and its application for pose-invariant face recognition system," in *Proc. IEEE Int. Conf. on Automatic Face and Gesture Recognition*, 2000, pp. 142–149.

[44] K. Okada and C. von der Malsburg, "Pose-invariant face recognition with parametric linear subspaces," in *Proc. IEEE Int. Conf. on Automatic Face and Gesture Recognition*, 2002, pp. 64–69.

[45] C. Sanderson, S. Bengio, and Y. Gao, "On transforming statistical models for non-frontal face verification," *Pattern Recogn.*, vol. 39, no. 2, pp. 288–302, 2006.

[46] D. Gonzalez-Jimenez, F. Sukno, J. L. Alba-Castro, and A. F. Frangi, "Automatic pose correction for local feature-based face authentication," in *Proc. IEEE Workshop on Motion of Non-Rigid and Articulated Objects, LNCS vol. 4069*, 2006, pp. 356–365.

[47] T. F. Cootes, G. V. Wheeler, K. N. Walker, and C. J. Taylor, "View-based active appearance models," *Image Vis. Comput.*, vol. 20, pp. 657–664, 2002.

[48] R. Gross, I. Matthews, and S. Baker, "Active Appearance Models with occlusion," *Image Vis. Comput.*, vol. 24, no. 6, pp. 593–604, 2006.

[49] K.-W. Wan, K.-M. Lam, and K.-C. Ng, "An accurate active shape model for facial feature extraction," *Pattern Recogn. Lett.*, vol. 26, no. 15, pp. 2409–2423, 2005.

[50] B. F. Buxton and M. B. Dias, "Implicit, view invariant, linear flexible shape modelling," *Pattern Recogn. Lett.*, vol. 26, no. 4, pp. 433–447, 2005.

[51] Y. Zhou, W. Zhang, X. Tang, and H. Shum, "A bayesian mixture model for multi-view face alignment," in *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, vol. 2, 2005, pp. 741–746.

[52] C. Butakoff and A. F. Frangi, "A framework for weighted fusion of multiple statistical models of shape and appearance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1847–1857, 2006.

[53] R. P. Brent, *Algorithms for Minimization Without Derivatives*. Prentice-Hall, 1973.

[54] A. Ortega, F. Sukno, E. Lleida, A. F. Frangi, A. Miguel, L. Buera, and E. Zacur, "AV@CAR: A spanish multichannel multimodal corpus for in-vehicle automatic audio-visual speech recognition," in *Proc. Int. Conf. on Language Resources and Evaluation*, vol. 3, 2004, pp. 763–767.

[55] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2004.

[56] C. Huang, H. Ai, Y. Li, and S. Lao, "High-performance rotation invariant multiview face detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 4, pp. 671–686, 2007.

[57] C. Hu, R. Feris, and M. Turk, "Real-time view-based face alignment using active wavelet networks," in *Proc. IEEE Int. Workshop on Analysis and Modeling of Faces and Gestures*, 2003, pp. 215 – 221.

[58] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Proc. IEEE Workshop on Applications of Computer Vision*, 1994, pp. 138–142.

[59] W. Gao, B. Cao, S. Shan, D. Zhou, X. Zhang, and D. Zhao, "The CAS-PEAL large-scale chinese face database and baseline evaluations," ICT-ISVISION Joint Research&Development Laboratory for Face Recognition, Chinese Academy of Sciences, Tech. Rep. JDL-TR-04-FR-001, 2004.

[60] J. Woo, "A short history of the development of 3-D ultrasound in obstetrics and gynecology," April 2009. [Online]. Available: http://www.ob-ultrasound.net/history-3D.html

[61] N. P. Nikitin, C. Constantin, P. H. Loh, J. Ghosh, E. I. Lukaschuk, A. Bennett, S. Hurren, F. Alamgir, A. L. Clark, and J. G. Cleland, "New generation 3-dimensional echocardiography for left ventricular volumetric and functional measurements: comparison with cardiac magnetic resonance." *Eur. J. Echocardiogr.*, vol. 7, no. 5, pp. 365–372, 2006.

[62] N. G. Bellenger, M. I. Burgess, S. G. Ray, A. Lahiri, A. J. S. Coats, J. G. F. Cleland, and D. J. Pennell, "Comparison of left ventricular ejection fraction and volumes in heart failure by echocardiography, radionuclide ventriculography and cardiovascular magnetic resonance. Are they interchangeable?" *Eur. Heart J.*, vol. 21, pp. 1387–1396, 2000.

[63] J. A. Noble and D. Boukerroui, "Ultrasound image segmentation: a survey," *IEEE Trans. Med. Imag.*, vol. 25, no. 8, pp. 987–1010, 2006.

[64] A. F. Frangi, W. J. Niessen, M. A. Viergever, and B. P. F. Lelieveldt, "A survey of three-dimensional modeling techniques for quantitative functional analysis of cardiac images," in *Advanced Image Processing in Magnetic Resonance Imaging*, L. Landini, V. Positano, and M. F. Santarelli, Eds. CRC Press, 2005.

[65] B. P. F. Lelieveldt, A. F. Frangi, S. C. Mitchell, H. C. van Assen, S. Ordas, J. H. C. Reiber, and M. Sonka, "3D active shape and appearance models in medical image analysis," in *Handbook of Mathematical Models of Computer Vision*, O. Faugeras, N. Paragios, and Y. Chen, Eds. Springer, 2006, pp. 471–486.

[66] E. D. Angelini, S. Homma, G. Pearson, J. W. Holmes, and A. F. Laine, "Segmentation of real-time three-dimensional ultrasound for quantification of ventricular function: A clinical study on right and left ventricles," *Ultrasound Med. Biol.*, vol. 31, no. 9, pp. 1143–1158, 2005.

[67] E. D. Angelini, Y. Jin, and A. F. Laine, "State-of-the-art of levelset methods in segmentation and registration of medical imaging modalities," in *Handbook of Biomedical Image Analysis. Registration Models*, J. S. Suri, D. Wilson, and S. Laxminarayan, Eds. Kluwer Academic/ Plenum Publishers, 2005, vol. 3.

[68] J. Montagnat and H. Delingette, "A review of deformable surfaces: topology, geometry and deformation," *Image Vis. Comput.*, vol. 19, no. 14, pp. 1023–1040, 2001.

[69] W. Hong, B. Georgescu, X. S. Zhou, S. Krishnan, Y. Ma, and D. Comaniciu, "Database-guided simultaneous multi-slice 3D segmentation for volumetric data," in *Proc. European Conf. on Computer Vision, LNCS vol. 3954*, 2006, pp. 397–409.

[70] V. Zagrodsky, V. Walimbe, C. R. Castro-Pareja, J. X. Qin, J.-M. Song, and R. Shekhar, "Registration-assisted segmentation of real-time 3-D echocardiographic data using deformable models," *IEEE Trans. Med. Imag.*, vol. 24, no. 9, pp. 1089–1099, 2005.

[71] Y. Ping, A. Sinusas, and J. S. Duncan, "LV segmentation from 3D echocardiography using fuzzy features and a multilevel FFD model," in *Proc. IEEE Int. Symp. on Biomedical Imaging*, 2007, pp. 848–851.

[72] M. Sonka, B. P. F. Lelieveldt, S. C. Mitchell, J. G. Bosch, R. J. van der Geest, and J. H. C. Reiber, "Active appearance motion model segmentation," in *Proc. USF Int. Workshop on Digital and Computational Video*, 2001, pp. 64–68.

[73] J. G. Bosch, S. C. Mitchell, B. P. F. Lelieveldt, F. Nijland, O. Kamp, M. Sonka, and J. H. C. Reiber, "Automatic segmentation of echocardiographic sequences by active appearance motion models," *IEEE Trans. Med. Imag.*, vol. 21, no. 11, pp. 1374–1383, 2002.

[74] J. Hansegård, S. Urheim, K. Lunde, and S. I. Rabben, "Constrained active appearance models for segmentation of triplane echocardiograms," *IEEE Trans. Med. Imag.*, vol. 26, no. 10, pp. 1391–1400, 2007.

[75] F. Orderud, J. Hansegård, and S. I. Rabben, "Real-time tracking of the left ventricle in 3D echocardiography using a state estimation approach," in *Proc. Int. Conf. Medical Image Computing and Computer Assisted Intervention, LNCS vol. 4791*, 2007, pp. 858–865.

[76] J. Hansegård, F. Orderud, and S. I. Rabben, "Real-time active shape models for segmentation of 3D cardiac ultrasound," in *Proc. Int. Conf. on Computer Analysis of Images and Patterns, LNCS vol. 4673*, 2007, pp. 157–164.

[77] F. Orderud, G. Kiss, and H. Torp, "Automatic coupled segmentation of endo- and epicardial borders in 3D echocardiography," in *Proc. IEEE Int. Ultrasonics Symp.*, 2008, pp. 1749–1752.

[78] J. Crosby, B. H. Amundsen, T. Hergum, E. W. Remme, S. Langeland, and H. Torp, "3-D speckle tracking for assessment of regional left ventricular function." *Ultrasound Med. Biol.*, vol. 35, no. 3, pp. 458–471, 2009.

[79] Q. Duan, E. D. Angelini, S. L. Herz, C. M. Ingrassia, K. D. Costa, J. W. Holmes, S. Homma, and A. F. Laine, "Region-based endocardium tracking on real-time three-dimensional ultrasound," *Ultrasound Med. Biol.*, vol. 35, no. 2, pp. 256–265, 2009.

[80] C. Corsi, G. Saracino, A. Sarti, and C. Lamberti, "Left ventricular volume estimation for real-time three-dimensional echocardiography," *IEEE Trans. Med. Imag.*, vol. 21, no. 9, pp. 1202–1208, 2002.

[81] S. Corsaro, K. Mikula, A. Sarti, and F. Sgallari, "Semi-implicit covolume method in 3D image segmentation," *SIAM J. Sci. Comput.*, vol. 28, no. 6, pp. 2248–2265, 2006.

[82] M. D. Cerqueira, N. J. Weissman, V. Dilsizian, A. K. Jacobs, S. Kaul, W. K. Laskey, D. J. Pennell, J. A. Rumberger, T. Ryan, and M. Verani, "Standardized myocardial segmentation and nomenclature for tomographic imaging of the heart: a statement for healthcare professionals from the Cardiac Imaging Committee of the Council on Clinical Cardiology of the American Heart Association," *Circulation*, vol. 105, no. 4, pp. 539–542, 2002.

[83] S. Ordas, E. Oubel, R. Leta, F. Carreras, and A. F. Frangi, "A statistical shape model of the heart and its application to model-based segmentation," in *Proc. SPIE*, vol. 6511, 2007.

[84] Y. Yu and S. T. Acton, "Speckle reducing anisotropic diffusion," *IEEE Trans. Image Process.*, vol. 11, no. 11, pp. 1260–1270, 2002.

[85] P. Abbott and M. Braun, "Simulation of ultrasound image data by a quadrature method," in *Eng. Phys. Sci. Med. Health Conf.*, vol. 209, 1996.

[86] J. C. Bamber and R. J. Dickinson, "Ultrasonic B-scanning: A computer simulation," *Phys. Med. Biol.*, vol. 25, pp. 463–479, 1980.

[87] J. A. Jensen and N. B. Svendsen, "Calculation of pressure fields from arbitrarily shaped, apodized, and excited ultrasound transducers," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 39, no. 2, pp. 262–267, 1992.

[88] J. A. Jensen, "Field: A program for simulating ultrasound systems," *Med. Biol. Eng. Comput.*, vol. 34, Supplement 1, Part 1, pp. 351–353, 1996.

[89] J. M. Bland and D. G. Altman, "Applying the right statistics: analyses of measurement studies," *Ultrasound Obstet. Gynecol.*, vol. 22, no. 1, pp. 85–93, 2003.

[90] J. Hung, R. Lang, F. Flachskampf, S. K. Shernan, M. L. McCulloch, D. B. Adams, J. Thomas, M. Vannan, T. Ryan, and A. S. E., "3d echocardiography: a review of the current status and future directions." *J. Am. Soc. Echocardiogr.*, vol. 20, no. 3, pp. 213–233, 2007.

[91] C. Jenkins, K. Bricknell, L. Hanekom, and T. H. Marwick, "Reproducibility and accuracy of echocardiographic measurements of left ventricular parameters using real-time three-dimensional echocardiography," *J. Am. Coll. Cardiol.*, vol. 44, no. 4, pp. 878–886, 2004.

[92] L. Sugeng, V. Mor-Avi, L. Weinert, J. Niel, C. Ebner, R. Steringer-Mascherbauer, F. Schmidt, C. Galuschky, G. Schummers, R. M. Lang, and H.-J. Nesser, "Quantitative assessment of left ventricular size and function: side-by-side comparison of real-time three-dimensional echocardiography and computed tomography with magnetic resonance reference," *Circulation*, vol. 114, no. 7, pp. 654–661, 2006.

[93] M. van Stralen, K. Y. E. Leung, M. M. Voormolen, N. de Jong, A. F. W. van der Steen, J. H. C. Reiber, and J. G. Bosch, "P2A-8 fully automatic detection of left ventricular long axis and mitral valve plane in 3D echocardiography," in *Proc. IEEE Int. Ultrasonics Symp.*, 2007, pp. 1488–1491.

[94] A. F. Frangi, W. J. Niessen, and M. A. Viergever, "Three-dimensional modeling for functional analysis of cardiac images, a review," *IEEE Trans. Med. Imag.*, vol. 20, no. 1, pp. 2–5, 2001.

[95] D. Shen, Y. Zhan, and C. Davatzikos, "Segmentation of prostate boundaries from ultrasound images using statistical shape model," *IEEE Trans. Med. Imag.*, vol. 22, no. 4, pp. 539–551, 2003.

[96] G. Subsol, J. P. Thirion, and N. Ayache, "A scheme for automatically building three-dimensional morphometric anatomical atlases: application to a skull atlas." *Med. Image Anal.*, vol. 2, no. 1, pp. 37–60, 1998.

[97] A. D. Brett and C. J. Taylor, "A method of automated landmark generation for automated 3D PDM construction," in *Proc. British Machine Vision Conf.*, vol. 2, 1998, pp. 914–923.

[98] A. F. Frangi, D. Rueckert, J. A. Schnabel, and W. J. Niessen, "Automatic construction of multiple-object three-dimensional statistical shape models: Application to cardiac modeling," *IEEE Trans. Med. Imag.*, vol. 21, no. 9, pp. 1151–1166, 2002.

[99] M. Kaus, V. Pekar, C. Lorenz, R. Truyen, S. Lobregt, and J. Weese, "Automated 3D PDM construction from segmented images using deformable models," *IEEE Trans. Med. Imag.*, vol. 22, no. 8, pp. 1005–1013, 2003.

[100] D. Rueckert, A. F. Frangi, and J. A. Schnabel, "Automatic construction of 3d statistical deformation models using non-rigid registration," *IEEE Trans. Med. Imag.*, vol. 22, no. 8, pp. 1014–1025, 2003.

[101] M. Ljungberg and S. E. Strand, "A Monte Carlo program for the simulation of scintillation camera characteristics." *Comput. Methods Programs Biomed.*, vol. 29, no. 4, pp. 257–272, 1989.

[102] T. K. Lewellen, R. Harrison, and S. Vannoy, "The SimSET program, in Monte Carlo calculations," in *Monte Carlo Calculations in Nuclear Medicine*, M. Ljungberg, S. Strand, and M. King, Eds. Philadelphia: Institute of Physics Publishing, 1998, pp. 77–92.

[103] OpenGate Collaboration, "Geant4 Application for Emission Tomography (GATE)." [Online]. Available: http://opengatecollaboration.healthgrid.org

[104] W. P. Segars, B. M. W. Tsui, E. C. Frey, and E. K. Fishman, "Extension of the 4D NCAT phantom to dynamic x-ray CT simulation," *IEEE Nuclear Science Symposium Conference Record*, vol. 5, pp. 3195–3199, 2003.

[105] K. M. Rosenberg, "The open source computed tomography simulator (CTSim)." [Online]. Available: http://www.ctsim.org

[106] H. Benoit-Cattin, G. Collewet, B. Belaroussi, H. Saint-Jalmes, and C. Odet, "The SIMRI project: A versatile and interactive MRI simulator," *J. Magn. Reson.*, vol. 173, no. 1, pp. 97–115, 2005.

[107] R. K. S. Kwan, A. C. Evans, and G. B. Pike, "MRI simulation-based evaluation of image-processing and classification methods," *IEEE Trans. Med. Imag.*, vol. 18, no. 11, pp. 1085–1097, 1999.

[108] G. Germano and D. S. Berman, "Quantitative gated perfusion SPECT," in *Clinical gated cardiac SPECT*. Futura Publishing Co, NY, 1999, pp. 115–146.

[109] G. Germano, H. Kiat, B. Kavanagh, M. Moriel, M. Mazzanti, H. Su, K. F. Van Train, and D. S. Berman, "Automatic quantification of ejection fraction from gated myocardial perfusion SPECT," *J. Nucl. Med.*, vol. 36, no. 11, pp. 2138–2147, 1995.

[110] T. L. Faber, C. Cooke, R. Folks, J. Vansant, K. N. E. DePuey, R. Pettigrew, and E. V. Garcia, "Left ventricular function and perfusion from gated SPECT perfusion images: An intergrated method," *J. Nucl. Med.*, vol. 40, no. 4, pp. 650–659, 1999.

[111] L. Stegger, C. S. Lipke, P. Kies, B. Nowak, O. Schober, U. Buell, M. Schafers, and W. M. Schaefer, "Quantification of left ventricular volumes and ejection fraction from gated 99mTc-MIBI SPECT: validation of an elastic surface model approach in comparison to cardiac magnetic resonance imaging, 4D-MSPECT and QGS," *Eur. J. Nucl. Med. Mol. Imaging*, vol. 34, no. 6, pp. 900–909, 2007.

[112] A. D. Achtert, M. A. King, S. T. Dahlberg, P. H. Pretorius, K. J. LaCroix, and B. M. W. Tsui, "An investigation of the estimation of ejection fractions and cardiac volumes by a quantitative gated SPECT software package in simulated gated SPECT images," *J. Nucl. Cardiol.*, vol. 5, no. 2, pp. 144–52, 1998.

[113] E. Vallejo, D. P. Dione, W. L. Bruni, R. T. Constable, P. P. Borek, J. P. Soares, J. G. Carr, S. G. Condos, F. J. T. Wackers, and A. Sinusas, "Reproducibility and accuracy of gated SPECT for determination of left ventricular volumes and ejection fraction: Experimental validation using MRI," *J. Nucl. Med.*, vol. 41, no. 5, pp. 874–882, 2000.

[114] J. Montagnat and H. Delingette, "4D deformable models with temporal constraints: application to 4D cardiac image segmentation." *Med. Image Anal.*, vol. 9, no. 1, pp. 87–100, 2005.

[115] D. Lingrand, A. Charnoz, P. M. Koulibaly, J. Darcourt, and J. Montagnat, "Toward accurate segmentation of the LV myocardium and chamber for volumes estimation in gated SPECT sequences," in *Proc. European Conf. on Computer Vision, LNCS vol. 3024*, 2004, pp. 267–278.

[116] X. He, E. C. Frey, J. M. Links, K. L. Gilland, W. P. Segars, and B. M. W. Tsui, "A mathematical observer study for the evaluation and optimization of compensation methods for myocardial SPECT using a phantom population that realistically models patient variability," *IEEE Trans. Nucl. Sci.*, vol. 51, no. 1, pp. 218–224, 2004.

[117] A. B. Barclay, R. L. Eisner, and E. V. DiBella, "Emory PET thorax model database," Carlyle Fraser Heart Center/Crawford Long Hospital of Emory University and Georgia Institute of Technology, Atlanta, GA, Tech. Rep., 1995. [Online]. Available: http://www.emory.edu/CRL/abb/thoraxmodel/contents.html

[118] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, 2000.

[119] K. J. LaCroix, B. M. W. Tsui, E. C. Frey, and R. Jaszczak, "Receiver operating characteristic evaluation of iterative reconstruction with attenuation correction in 99mTc-Sestamibi myocardial SPECT images," *J. Nucl. Med.*, vol. 41, no. 3, pp. 502–513, 2000.

[120] J. C. Gower, "Generalized procrustes analysis," *Psychometrika*, vol. 40, pp. 33–50, 1975.

[121] M. Ljungberg, A. Larsson, and L. Johansson, "A new collimator simulation in SIMIND based on the delta-scattering technique," *IEEE Trans. Nucl. Sci.*, vol. 52, no. 5, pp. 1370–1375, 2005.

[122] J. A. Fessler, "Users guide for ASPIRE 3D image reconstruction software," Comm. and Sign. Proc. Lab., Dept. of EECS, Univ. of Michigan, Ann Arbor, MI, Tech. Rep. 310, 1997.

[123] C. Tobon-Gomez, C. Butakoff, S. Ordas, S. Aguade, and A. F. Frangi, "Comparative study of diversely trained 3D-ASM models for segmentation of gated SPECT data," in *Proc. SPIE*, vol. 6511, 2007.

[124] W. P. Segars, "Development and application of the new dynamic nurbs-based cardiac-torso (NCAT) phantom," Ph.D. dissertation, The University of North Carolina, 2001.

[125] J. D. Gibbons, *Nonparametric Statistical Inference*, 2nd ed. Marcel DEkker, Inc, 1985.

[126] C. Vanhove, P. R. Franken, M. Defrise, A. Momen, H. Everaert, and A. Bossuyt, "Automatic determination of left ventricular ejection fraction from gated blood-pool tomography," *J. Nucl. Med.*, vol. 42, no. 3, pp. 401–407, 2001.

[127] J. M. Bland and D. G. Altman, "Statistical methods for assessing agreement between two methods of clinical measurement." *Lancet*, vol. 1, no. 8476, pp. 307–310, 1986.

[128] J. Bogaert, "Cardiac function," in *Clinical Cardiac MRI*, J. Bogaert, S. Dymarkowski and A.M. Taylor, Ed., vol. 1. Springer-Verlag New York, Inc, 2005, pp. 99–134.

[129] E. Tadamura, T. Kudoh, M. Motooka, M. Inubushi, T. Okada, S. Kubo, N. Hattori, T. Matsuda, T. Koshiji, K. Nishimura, M. Komeda, and J. Konishi, "Use of technetium-99m sestamibi ECG-gated single-photon emission tomography for the evaluation of left ventricular function following coronary artery bypass graft: comparison with threedimensional magnetic resonance imaging," *Eur. J. Nucl. Med.*, vol. 26, no. 7, pp. 705–712, 1999.

[130] C. D. L. Bavelaar-Croon, H. W. M. Kayser, E. E. van der Wall, A. de Roos, P. Dibbets-Schneider, E. K. J. Pauwels, G. Germano, and D. E. Atsma, "Left ventricular function: Correlation of quantitative gated SPECT and MR imaging over a wide range of values," *Radiology*, vol. 217, pp. 572–575, 2000.

[131] K. Nakajima, T. Higuchi, J. Taki, M. Kawano, and N. Tonami, "Accuracy of ventricular volume and ejection fraction measured by gated myocardial SPECT: comparison of 4 software programs," *J. Nucl. Med.*, vol. 42, no. 10, pp. 1571–1578, 2001.

[132] C. S. A. Lipke, H. P. Kuhl, B. Nowak, H. J. Kaiser, P. Reinartz, K. C. Koch, U. Buell, and W. Schaefer, "Validation of 4D-MSPECT and QGS for quantification of left ventricular volumes and ejection fraction from gated 99mTc-MIBI SPET: comparison with cardiac magnetic resonance imaging," *Eur. J. Nucl. Med. Mol. Imaging*, vol. 31, no. 4, pp. 482–490, 2004.

[133] M. Lomsky, J. Richter, L. Johansson, H. El-Ali, K. Astrom, M. Ljungberg, and L. Edenbrandt, "A new automated method for analysis of gated SPECT images based on a three-dimensional heart shaped model," *Clin. Physiol. Funct. Imaging*, vol. 25, no. 4, pp. 234–240, 2005.

[134] W. M. Schaefer, C. S. A. Lipke, D. Standke, H. P. Kuhl, B. Nowak, H. J. Kaiser, K. C. Koch, and U. Buell, "Quantification of left ventricular volumes and ejection fraction from gated 99mTc-MIBI SPECT: MRI validation and comparison of the emory cardiac tool box with QGS and 4D-MSPECT," *J. Nucl. Med.*, vol. 46, no. 8, pp. 1256–1263, 2005.

[135] Y. W. Wu, E. Tadamura, M. Yamamuro, S. Kanao, S. Okayama, N. Ozasa, M. Toma, T. Kimura, M. Komeda, and K. Togashi, "Estimation of global and regional cardiac function using 64-slice computed tomography: A comprasion study with echocardiography, gated-SPECT and cardiovascular magnetic resonance," *Int. J. Cardiol.*, vol. 128, no. 1, pp. 69–76, 2008.

[136] J. P. A. Ioannidis, T. A. Trikalinos, and P. G. Danias, "Electrocardiogram-gated single-photonemission computed tomography versus cardiacmagnetic resonance imaging for the assessmentof left ventricular volumes and ejection fraction: A meta-analysis," *J. Am. Coll. Cardiol.*, vol. 39, no. 12, pp. 2059–2068, 2002.

[137] J. A. Case, S. J. Cullom, T. M. Bateman, C. Bamhart, and M. Saunders, "Overestimation of LVEF by gated MIBI myocardial perfusion SPECT in patients with small hearts," *J. Am. Coll. Cardiol.*, vol. 31, no. 1, pp. 43–43, 1998.

[138] C. Butakoff, S. Balocco, S. Ordas, and A. F. Frangi, "Simulated 3D ultrasound LV cardiac images for active shape model training," in *Proc. SPIE*, vol. 6512, 2007.

Publications

Within the scope of the thesis

Journals

- C. Butakoff, S. Balocco, F. M. Sukno, C. Hoogendoorn, C. T. Gomez, G. Avegliano, A. F. Frangi, "Left-ventricular Epi- and Endocardium Extraction from 3D Ultrasound Using an Automatically Constructed 3D ASM", under review.
- C. Butakoff and A. F. Frangi, "Multi-View Face Segmentation Using Fusion of Statistical Shape and Appearance Models", under review.
- C. Tobon-Gomez, C. Butakoff, S. Ordas, S. Aguade, and A.F. Frangi, "Automatic Construction of 3D-ASM Intensity Models by Simulating Image Acquisition: Application to Myocardial Gated SPECT Studies", **IEEE Transactions on Medical Imaging** 27(11):1655-1667, 2008.
- C. Butakoff and A. F. Frangi, "A Framework for Weighted Fusion of Multiple Statistical Models of Shape and Appearance", **IEEE Transactions On Pattern Analysis and Machine Intelligence** 28(11):1847-1857, 2006.

Conferences

- C. Butakoff, F. M. Sukno, B. H. Bijnens, A. F. Frangi, Robust LV sequence segmentation from 3D echocardiographic images, submitted.
- C. Butakoff, S. Balocco, S. Ordas, A. F. Frangi, "Simulated 3D Ultrasound LV Cardiac Images for Active Shape Model Training", **SPIE Conference on Medical Imaging** 6512, 2007
- C. Tobon-Gomez, C. Butakoff, S. Ordas, A. F. Frangi, "Comparative Study of Diverse Model Building Strategies for 3D-ASM Segmentation of Dynamic Gated SPECT Data", **SPIE Conference on Medical Imaging** 6511, 2007

Outside the scope of the thesis

Journals

- I. Aizenberg and C. Butakoff, "A Windowed Gaussian Notch Filter for Quasi-Periodic Noise Removal", **Image and Vision Computing** 26(10):1347-1353, 2008
- D. Delgado, C. Butakoff, B. Ersboll, W. Stoecker, "Independent Histogram Pursuit for Segmentation of Skin Lesions", **IEEE Transactions on Biomedical Engineering** 55(1): 157-161, 2008
- D. Delgado, C. Butakoff, B. Ersboll, J. Carstensen, "Automatic Change Detection and Quantifcation of Dermatological Diseases with an Application to Psoriasis Images", **Pattern Recognition Letters** 28(9): 1012-1018, 2007
- F. Sukno, S. Ordas, C. Butakoff, S. Cruz and A. F. Frangi, "Active Shape Models With Invariant Optimal Features: Application to Facial Analysis", **IEEE Transactions On Pattern Analysis and Machine Intelligence** 29(7):1105-1117, 2007.
- I. Aizenberg and C. Butakoff, "Impulsive Noise Removal using Threshold Boolean Filtering based on the Impulse Detecting Functions", **IEEE Signal Processing Letters** 12(1):63-66, 2005.
- I. Aizenberg and C. Butakoff, "Effective Impulse Detectors Based on Rank-Order Criteria", **IEEE Signal Processing Letters** 11(3):363- 366, 2004.
- I. Aizenberg and C. Butakoff, "Superresolution and Supersampling by Spectral Extrapolation", **Pattern Recognition and Image Analysis** 14(3): 370-379, 2004.

Conferences

- F. M. Sukno, S.-K. Pavani, C. Butakoff and A. F. Frangi, "Automatic Assessment of Eye Blinking Patterns through Statistical Shape Models", submitted.
- I. Larrabide, M. Nieber, X. Planes, J. A. Moya, C. Butakoff, R. Sebastian, O. Camara, M. De Craene, B. H. Bijnens, and A. F. Frangi, "GIMIAS: An open source framework for efficient development of research tools and clinical prototypes", **Functional Imaging and Modeling of the Heart**, LNCS 5528, pp. 417-426, 2009.
- S.-K. Pavani, F. M. Sukno, C. Butakoff, X. Planes and A. F. Frangi, "A Confidence-Based Update Rule for Self-updating Human Face Recognition Systems", **IAPR/IEEE International Conference on Biometrics**, LNCS 5558, pp. 151-160, 2009.
- F. Sukno, A.Frangi, S. Cruz, S. Ordas, and C. Butakoff, "Active Shape Models with Invariant Optimal Features (IOF-ASM)", **Conference on Audio- and Video-based Biometric Person Authentication**, LNCS 3546, pp. 365-375, 2005

Resumen

La presente tesis se centra en los aspectos de construcción y combinación de modelos activos de forma y de apariencia. Estos modelos representan una herramienta ampliamente utilizada para la segmentación y modelado de objetos mediante restricciones de forma y textura basadas en la estadística aprendida de un conjunto de entrenamiento. Sin embargo, estos modelos tienen varios problemas:

1. el costoso proceso de entrenamiento (tanto en relación al tiempo necesario como a la demanda de memoria);
2. la necesidad de un conjunto de entrenamiento de imágenes con el objeto delineado (por lo general de forma manual);
3. el alto grado de incertidumbre de dichos delineados (por la presencia de ruido), e incluso la imposibilidad práctica de realizar demarcaciones manuales cuando se trabaja en tres dimensiones.

Para resolver estos problemas se propone:

1. Un framework para la fusión ponderada de varios modelos activos de forma y apariencia basado en la combinación de autoespacios. Esta estrategia de combinación puede ser entendida como una interpolación lineal de los modelos. El modelo fusionado permite segmentar los objetos cuya apariencia se puede representar aproximadamente como una combinación lineal de los objetos que corresponden a los modelos fusionados. En otras palabras, si un objeto tiene una serie de apariciones típicas (diferentes expresiones o vistas faciales o diferentes patologías cardiacas), es posible elegir las apariciones más representativas y asumir que cualquier otra es una combinación lineal de dicho conjunto representativo. De este modo, la fusión de modelos se puede utilizar para segmentar la imagen y los pesos de la combinación pueden ser utilizados para determinar cual modelo representa mejor al objeto. Las posibles aplicaciones de este framework son: construcción incremental del modelo, clasificación basada en los pesos de la combinación y reducción del conjunto de entrenamiento hasta tener solo las apariencias características.

2. Un algoritmo de segmentación de caras de varias vistas basado en la fusión de modelos activos de apariencia. Este algoritmo se utiliza para la segmentación de cualquier vista facial y también para determinar el ángulo de la vista a partir de los pesos de la fusión. Se construyen sólo los modelos que corresponden a las vistas extremas y se supone que el resto de las vistas son combinación lineal de las extremas. Estimación de los pesos de fusión mediante la minimización de error de reconstrucción permite encontrar el modelo combinado óptimo que mejor se adapta a la imagen segmentada.
3. La combinación de imágenes de tomografía computada (CT), ultrasonido (US) y tomografía computarizada por emisión de fotones individuales (SPECT) para crear automaticamente un modelo activo de forma. Se demuestra cómo la estadística de la apariencia puede ser aprendida para dos modalidades donde la resolución o calidad son demasiado bajas para obtener marcaciones manuales fiables de los contornos del objeto, especialmente en 3D. En este caso la generación de imágenes sintéticas, a través de la simulación del proceso de formación de la imagen, permite sintetizar un conjunto de entrenamiento para la apariencia a partir de un conjunto de formas obtenidas de las imágenes de alta calidad que ofrece CT.

Acknowledgements

I would like to gratefully acknowledge the patient supervision of Dr. Alejandro F. Frangi during this work. On the other hand I would like to thank my friends and colleagues Dr. Federico M. Sukno and Kaushik Pavani, who contributed with their ideas and criticisms, and of course all the people from both Universidad de Zaragoza and Universitat Pompeu Fabra I had pleasure collaborating with, who created this unique atmosphere, which made this work possible. Special thanks go to my friend Dr. Igor Aizenberg, who opened for me the doors to the world of science, and to my parents.

This page was intentionally left blank.