Parametric Optimization of a Model-Based Segmentation Algorithm for Cardiac MR Image Analysis: A Grid-Computing Approach

S. ORDAS\textsuperscript{a,1}, H.C. VAN ASSEN\textsuperscript{b}, J. PUENTE\textsuperscript{c}, B.P.F. LELIEVELDT\textsuperscript{b} and A.F. FRANGI\textsuperscript{a}

\textsuperscript{a}Computational Imaging Laboratory, Universitat Pompeu Fabra, Barcelona, Spain
\textsuperscript{b}Division of Image Processing, Department of Radiology, Leiden University Medical Center, Leiden, The Netherlands
\textsuperscript{c}GridSystems, S.A., Mallorca, Spain

Abstract. In this work we present a Grid-based optimization approach performed on a set of parameters that affects both the geometric and grey-level appearance properties of a three-dimensional model-based algorithm for cardiac MRI segmentation. The search for optimal values was assessed by a Monte Carlo procedure using computational Grid technology. A series of segmentation runs were conducted on an evaluation database comprising 30 studies at two phases of the cardiac cycle (60 datasets), using three shape models constructed by different methods. For each of these model-patient combinations, six parameters were optimized in two steps: those which affect the grey-level properties of the algorithm first and those relating to the geometrical properties, secondly. Two post-processing tasks (one for each stage) collected and processed (in total) more than 70000 retrieved result files. Qualitative and quantitative validation of the fitting results indicates that the segmentation performance was greatly improved with the tuning. Based on the experienced benefits with the use of our middleware, and foreseeing the advent of large-scale tests and applications in cardiovascular imaging, we strongly believe that the use of Grid computing technology in medical image analysis constitutes a real necessity.

Keywords. Parametric optimization, Grid computing, Statistical model-based segmentation, MRI analysis

Introduction

In the last few years, many model-based approaches for image segmentation have contributed to the quite evolving field of medical image analysis. The rationale behind these methods is to analyze the image in a top-down fashion: a generic template model of the structure of interest is subsequently instantiated and deformed to accommodate for the clues provided by image information. For a cardiac application, it is possible to
learn the shape statistics of a heart from a population of healthy and/or diseased hearts, and construct a compact and specific anatomical model. Statistical model-based approaches work in this way, and are able to provide constraints that allow for efficiently handling situations with substantial sparse or missing information. Statistical models of shape (ASM) [1] and appearance (AAM) [2] variability are two model-driven segmentation schemes very popular in medical image analysis. In building statistical models, a set of segmentations of the shape of interest is required, as well as a set of corresponding landmarks defined over them. An ASM comprises a shape and an appearance model. The former primarily holds information about the shape and its allowed variations in a Point Distribution Model (PDM), determined by a Principal Component Analysis (PCA). The latter is responsible of learning grey-level patterns from the training set image data, which are to be compared against those identified in a new (unknown) image during the fitting stage. The algorithm therefore consists of an iterative process in which the appearance model looks into the image for new candidate positions to deform the shape model, and the shape model applies statistical geometric constraints in order to keep the deformation process always within legal statistical limits. ASMs and AAMs are currently becoming a popular topic of research towards three-dimensional (3D) cardiac image segmentation. First approaches like [3,4] and [5,6] have evidenced an encouraging performance in segmenting the left ventricle in 3D datasets of MR/US and MR/CT, respectively.

In the work that we present here, we describe the methodology employed to find the optimal set of parameters for the 3D-ASM segmentation algorithm of van Assen et al. [6] that uses a fuzzy inference system in the appearance model. Moreover, by running the segmentation tests with three different shape models, we aimed to explore to what extent the use of our method for automatically building shape models [7,8] does really improve the segmentation performance of a model-based fitting approach. The way our Grid middleware is designed to work allowed for quite an easy and general methodology for setting up, running, and post-processing the results of the parametric optimization. After presenting the general ideas behind the algorithm, we provide some comments on our experience with the use of Grid computing, and conclude the paper with a discussion.

1. Active Shape Modeling

1.1. Shape Model

Consider a set \( X = \{x_i; i = 1...n\} \) of \( n \) shapes. Each shape is described by the concatenation of \( m \) 3D landmarks \( p_j = \{p_{1j}, p_{2j}, p_{3j}\} \), \( j = 1...m \) obtained from a surface triangulation. \( X \) is thus a distribution in a \( 3m \)-dimensional space. This representation allows approximating any shape by using the following linear model

\[
x = \bar{x} + \Phi b
\]  

where \( \bar{x} \) is the average landmark vector, \( b \) is the shape parameter vector of the model, and \( \Phi \) is a matrix whose columns are the principal components of the covari-
The principal components of $S$ are the eigenvectors, $\phi_i$, with corresponding eigenvalues, $\lambda_i$ (sorted so that $\lambda_i \geq \lambda_{i+1}$). If $\Phi$ contains only the first $t < \min\{m,n\}$ eigenvectors corresponding to the largest non-zero eigenvalues, we can approximate any shape of the training set $\mathcal{X}$, using Equation 1; where $\Phi = \{\phi_1, \phi_2, \ldots, \phi_t\}$ and $b$ is a $t$-dimensional vector given by $b = \Phi^T (x - \bar{x})$. Assuming that the cloud of landmark vectors follows a multi-dimensional Gaussian distribution, the variance of the $i$-th parameter $b_i$ across the training set is given by $\lambda_i$. The samples are normalized with respect to a reference coordinate frame, eliminating differences across objects due to rotation, translation and size. Once the shape samples are aligned, the remaining differences are solely shape related. By varying the model parameters $b_i$ in Equation 1, different instances of the shape class under analysis will result. Finally, by applying limits to the variation of $b_i$,

$$|b_i| \leq \beta \sqrt{\lambda_i} \quad (2)$$

with $\beta$ usually less than 3, it can be enforced that a generated shape is similar to the shapes contained in the training set. See Figure 1 for some examples of valid shapes generated by the model.

1.2. Appearance Model

In order to deform the shape model, candidate points are collected from the image data using a decision scheme based on the Takagi-Sugeno Fuzzy Inference System (FIS) [9]. For a complete description of this decision scheme, we refer to [6]. The Fuzzy C-Means (FCM) within the FIS yields three membership distributions for the tissues: air, myocardium and blood. In the inference part of the FIS, crisp values per
pixel are derived based on the fuzzy membership distributions. This is performed using two kinds of thresholds (see Figure 2):

1. **Gray level threshold**: first, a threshold is placed in the membership distributions, marking a part that is attributed to a tissue class irrespective of its membership value for that class. All pixels with a gray value below the threshold are assigned to the air-class. The position of the threshold is determined as a proportion between tissue class centers. The threshold is placed at a preset proportion between the tissue class centers of the air and the myocardium tissue classes, resulting from the FCM.

2. **Membership degree thresholds**: the gray level threshold above divides the membership distributions from the FCM in two parts, assigning tissue labels to the lower part. The remaining part is classified using membership threshold levels. Thus, pixels with a gray value in this part are assigned the tissue label of the class with the highest membership value at that particular gray value, provided this membership value is above the threshold for the tissue. Pixels with gray values whose highest membership value does not reach the corresponding tissue membership threshold are left unclassified.

### 1.3. Using Sectors in FCM

In the appearance model, the FCM fits a number of (Gaussian) gray value distributions to a given combined histogram of image patches extracted from the studied dataset. To ensure a large enough population of gray values for a stable application of FCM, all patches from all intersections of the model mesh with the image slices are collected. However, in many datasets, intensity inhomogeneity was observed within tissues. These effects can severely hamper the classification of the tissues based on sampled gray value histograms. To prevent this, the shape model was organized in sectors, by assignment of labels. Thus, different sectors can be assigned different rule sets in the FIS, and (combinations of) different sectors can be clustered in separate FCM operations. Since intensity inhomogeneity within one sector is limited, effects of inhomoge-
neity on the fuzzy clustering outcome are diminished. In total, seven sectors were defined on the two surfaces of the model (see Figure 3). After grouping of sectors with approximately the same level of inhomogeneity, five groups remained, leading to five different FCM operations to cover the whole cardiac shape. Table 1 shows the combinations of sectors with respect to FCM operations.

2. Matching Procedure

In summary, the 3D-ASM matching can be described as follows. The mean shape of the geometrical model is placed in the target dataset with a given initial scale and poses. The image planes intersect the model’s sub-part surface meshes (two in our case), yielding stacks of 2D contours. These contours are composed of the intersections of the image planes with individual mesh triangles. Candidate displacements (update-vectors) are propagated to the nodes in the mesh (see Figure 3). To facilitate through-plane motion in the model, they are projected on the normals of the model mesh, which also have a component perpendicular to the imaging planes. The current model state is aligned to the proposed model state resulting from the model update information using the method of Besl and McKay [10] for 3D point clouds, effectively eliminating scaling, translation and rotation differences. The residual shape differences are projected on the shape model subspace, yielding a model parameter vector. Thus, the proposed shape is approximated by the shape model using Equation 1, within statistical limits set beforehand by Equation 2. The steps above are repeated either for a fixed number of iterations, or until convergence is achieved.

3. Parametric Optimization

3.1. Parameters Related to the Shape Model

For every intersection of the model mesh with an image plane, a model update is computed as explained before. For every single update, the possibility exists that a no optimal or even an erroneous position results. To diminish the effects of erroneous updates, the update itself, which acts as a force on a model surface, is smeared out over a small region on the model surface around the source location. Thus, faulty or less reliable updates can be compensated for by a number of neighboring updates. The contribution of a single model update to other updates is weighted with a Gaussian kernel, with the actual weight depending on the geodesic distance along the mesh edges to the source

<table>
<thead>
<tr>
<th>FCM operation</th>
<th>Sectors</th>
<th>Anatomical Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1,2,3</td>
<td>Epicardial lateral wall</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>Epicardial RV wall</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>Epicardial apex</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>Endocardial lateral wall</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>Endocardial septum</td>
</tr>
</tbody>
</table>
location of the update. To limit the extent of the propagation of the updates, propaga-
tion is stopped either after a fixed number of edges, or when the weight attributed to a
particular update is below a preset threshold (see Figure 3(b)). The actual values for
the standard deviation of the kernel \( \sigma \), the propagation level limit \( \chi \), and
the number of standard deviations \( \beta \) that each shape model parameter is
allowed to vary (Equation 2), are the three shape-related parameters to optimize (see
Table 2).

### 3.2. Parameters Related to the Appearance Model

The three membership thresholds (horizontal in Figure 2) mentioned in Section 1.2 (for
air, myocardium, and blood) constitute the three parameters to update \( t_1, t_2, t_3 \) in
relationship with the appearance model. Their tuning ranges are specified in Table 2.

![Figure 3. Shape model sectors (left): (1-3) epicardial lateral wall, (4) epicardial RV wall, (5) epicardial apex,
(6) endocardial lateral wall, (7) endocardial RV wall. Updated propagation from a single source (right) with
the extent limited by a Gaussian kernel.](image)

<table>
<thead>
<tr>
<th>Tissue</th>
<th>Lower</th>
<th>Upper</th>
<th>Step Size</th>
<th>ED Optimal</th>
<th>ES Optimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appearance model parameters</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blood</td>
<td>0.1</td>
<td>0.5</td>
<td>0.1</td>
<td>0.2</td>
<td>0.4</td>
</tr>
<tr>
<td>Myocardium</td>
<td>0.05</td>
<td>0.3</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>Air</td>
<td>0.3</td>
<td>0.7</td>
<td>0.1</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>Shape model parameters</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sigma, ( \sigma )</td>
<td>3</td>
<td>9</td>
<td>1</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>Extent, ( \chi )</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Beta, ( \beta )</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
3.3. Fixed Settings

The 3D-ASM was set to run for a fixed number of iterations ($N=100$) using 60 modes of variation (more than 95% of the total variance of the three shape models tested). Always, the shape resulting from the final iteration was taken for assessment of its point-to-surface (P2S) error with respect to its manual segmented counterpart. The optimal settings for the algorithm where chosen based on the unsigned P2S distance measures averaged over the complete evaluation database.

3.4. Evaluation Dataset

The dataset used for the segmentation tests comprised 30 subjects. Fifteen were short axis scans of healthy volunteers acquired at the Leiden University Medical Center (Leiden, The Netherlands) using the balanced FFE-protocol on a Philips Gyroscan NT Intera, 1.5 T MR scanner (Philips Medical Systems, Best, Netherlands). The slice thickness was $8\, \text{mm}$, with a slice gap of $2\, \text{mm}$. The in-plane pixel resolution was $1.36 \times 1.36\, \text{mm}^2$. The other fifteen studies corresponded to patients from CETIR Sant Jordi Cardiovascular MR Centre (Barcelona, Spain) acquired using a General Electric Signa CV/i, 1.5 T scanner (General Electric, Milwaukee, USA) with the FIESTA scan protocol. The slice thickness was $8–10\, \text{mm}$ with an in-plane pixel resolution of $1.56 \times 1.56\, \text{mm}^2$. Selected patients suffered from common cardiac diseases: myocardium infarction (10), hypertrophy (2), and pericarditis (3). Expert segmentations were manually drawn along the epicardial ($LV$-epi) and endocardial ($LV$-endo) borders of the left ventricle ($LV$), at two phases of the cardiac cycle: End Systole (ES) and End Diastole (ED). Manual segmentations usually have an average intra- and inter-observer variability in the range of 1–2 mm. Nevertheless, they generally constitute the gold standard for assessing the achieved accuracy.

4. Grid Computing Approach

Our Grid middleware platform is the InnerGrid Nitya (GridSystems, Mallorca, Spain). An important aspect of this framework is that no special knowledge about Grid computing or particular hardware is required. This allows for easily setting up the system.
on a federation of off-the-shelf computers. The deployment of applications or services is usually straightforward and only requires some programming skills if it is intended to include the Grid in a more complicated workflow, comprising both distributed and non-distributed parts. The main components of the system are the Server, which is the coordinator of the Grid; the Agents, which are the pieces of software installed in all the machines federated in the Grid; and the Desktop Portal, a web graphical interface to the Grid and the applications running on it. Therefore the topology of the middleware is that of a Server connected to a large group of Agents that can be compiled for several platforms. When a service is requested, the Server looks at the information it receives from the Agents and decides which machines are able and idle enough to do the job related to that service. The Agents automatically receive all the elements they need, suitable for each operating system, and perform the requested task (which can be a part or the whole service). This includes any files, executables or parameters that might be needed. The Agents take care of managing all those components and perform their corresponding task. After everything is completed, the Server gathers the results and processes them. All this is hidden to the user and controlled by the Grid middleware.

The Grid service runs on a 45-node dual Xeon (2.8 GHz CPU, 2 GHz RAM) cluster, under Linux Red Hat 9, and is accessible through a web interface (Grid desktop). As a whole, the cluster represents more than 103 Gflops of computing power. As each node provides two Agents to the Grid, a total of 90 Agents were available. While all used nodes belong to the same computer cluster, the addition of nodes from other clusters is straightforward. For setting up the experiments, 60 datasets (30 patients at ED and ES) and 6 shape models (three construction methods and a different shape model for ED and ES) were uploaded to the server. The tuning of the parameters that affect the appearance model and the \( \beta \) parameter was run first. The remaining parameters were fixed at \( \chi = 3 \) and \( \sigma = 6 \) (values beforehand expected to be not far from optimal). The process produced approximately 64800 results and lasted 1.5 days. The optimal set of parameters of this first step was used in a second run for tuning the other two parameters that affect the shape model. This second run lasted 0.9 days and produced 5670 result files. In both runs, the post-processing was done in a local machine but could have been performed in the Grid server itself, as a post-processing task. Each of the collected result files was in fact a compressed folder (.tgz) comprising log files (with the segmentation errors) and a series of intermediate results and shape model instances, intended to track for events and visualization.

Figure 5. 3D-ASM segmentation. Initial (left) and final (right) states of the \( L V- e p i \) subpart. The triangulated shape corresponds to the fitted shape and the surfaced shape, to the manual one.
5. Quantitative Assessment

The performance assessment analysis was performed using the model state (i.e. position, orientation, scale and shape) from the last matching iteration of the fitting algorithm. Errors were measured by computing the mean unsigned P2S distances from regularly sampled points on the manually segmented contours to the surfaces of the fitted shapes. Two patient datasets were discarded from the assessment because their automatic segmentations were not comparable to the quality of the rest (for all models). The uncorrected field inhomogeneity in one case and a severe pericarditis on the other confounded the algorithm for any combination of the tuned parameters. Table 3 presents the automatic segmentation results, distinguishing between the shape model subparts (LV-endo and LV-epi). Values correspond to the shape model with best performance (in fact, the autolandmarked shape model presented in [7]). It is also indicated between brackets the percentage of improvement achieved by the optimization with respect to previously achieved segmentation accuracy using ad hoc settings. The third row corresponds to the error that the shape model would have for reconstructing the shapes (either ED or ES) if their exact positions were known. The automatic results are quite within clinically acceptable margins. Figure 4 shows an example of the initial and final model states in a single slice, while Figures 5 and 6 the same, but with the LV-endo and LV-epi subparts respectively. The manual shape is built from the manual contours and rendered as a surface. Figure 7 shows a typical result of the achieved segmentation.

6. Conclusions

The presented work serves as an application example of Grid computing in a medical image analysis task. Specifically, the exhaustive search of the optimal set of parameters for a three-dimensional model-based algorithm for cardiac MRI segmentation. The achieved improvement in the segmentation accuracy with respect to the use of parameters beforehand expected to be not far from optimal, was important (22–27.7% and 13–16.6% for ED and ES, respectively). From our experience with the middleware, we can
say that the set-up for distributing such a big number of tasks was a matter of minutes. The system took advantage of each resource depending on its usage, and not interfering with end users. When one of the computers in the Grid was not available, its tasks were automatically reassigned. Finally, the system collected the local output from all the units and made them available for download. A curious fact worth mentioning is that the whole procedure was done in The Netherlands using the Grid desktop (web portal), while the computer cluster was in Spain.

In conclusion, we believe that scientific progresses and derived clinical applications could be critically endangered by the lack of computational power or the inefficient scalability and expensiveness of traditional computational approaches. Grid technology solutions are quite valuable as they considerably shorten execution times of exhaustive searches and large-scale image processing, effectively enabling the shearing of computing resources between institutions. In the very active and evolving field of medical image analysis they have become a real necessity.

References
