AAM Segmentation of the Mandible and Brainstem

Release 0.00

Kola Babalola and Tim Cootes

July 20, 2009

Imaging Sciences Research Group
University of Manchester
Oxford Road
Manchester
M13 9PT
UK

Abstract

We describe an active appearance model (AAM) approach to the automatic segmentation of the mandible and brainstem. It involves four stages: Initialisation with a parts-and-geometry model, search with a global AAM followed by search with local AAMs, then post-processing using linear regressors.

Application of the method to the test images resulted in a mean (excluding subject 13) Dice overlap value of 76.1 $\pm$ 5.1% for the mandible, and 72.9 $\pm$ 24.6% for the brainstem. The method failed to segment both the mandible and brainstem in subject 13, and the whilst giving a successful segmentation of the mandible in subject 15, gave poor results on the brainstem. It takes about 17 minutes to run on a 64 bit Linux workstation with 2GB RAM, and a 2GHz pentium processor. We were encouraged by its performance on this dataset, and believe that its accuracy can be improved with some modifications to the segmentation pipeline.

Latest version available at the Insight Journal [http://hdl.handle.net/10380/3097]

Contents

1 Introduction

Appearance models capture the variation in shape and texture over a training set assumed to be representative of the class of object being modelled. Active appearance models (AAMs) introduced by Cootes *et al.*[2] are a method of using appearance models to locate instances of the object in images. AAMs have been used in a wide variety of applications from segmentation of medical images in 2D and 3D e.g. [9], [10] to face location in computer vision applications e.g. [5].

The approach taken in this paper is based on the framework of Babalola *et al.*[1] which involves constructing a global AAM of all the objects of interest, local AAMs of each individual object and linear regressors for detailed segmentation of each object. AAM search is a local optimisation method and therefore requires good initialisation. However, whilst [1] initialise their search by registration, we introduce of a novel initialisation method using a parts-and-geometry model. Figure 1 gives a schematic of our approach.

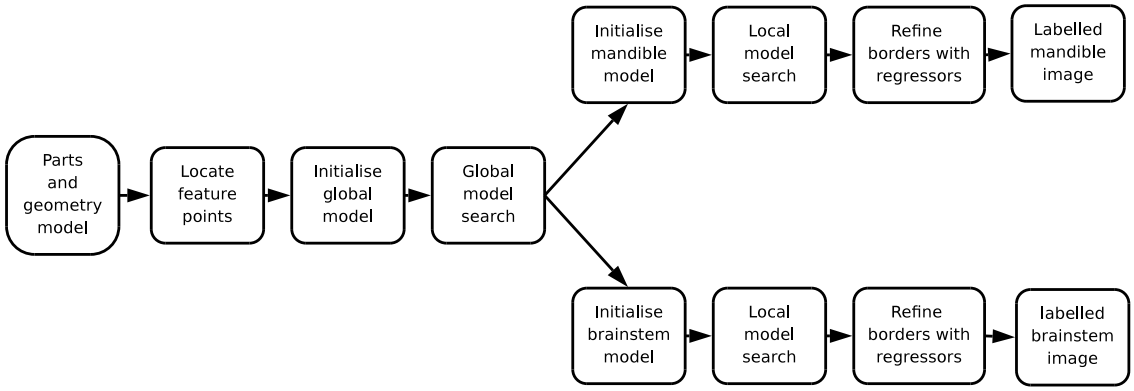

Figure 1: Outline of the stages involved in the segmentation

In the following we describe the construction of the parts-and-geometry models as they are the novel aspect of this work and give brief details of the construction of the AAMs and the regressors as they have been described elsewhere [1],[2]. We then show how the segmentation pipeline is applied and present results obtained on the test images provided in the challenge. We end with a discussion of the performance of our method.

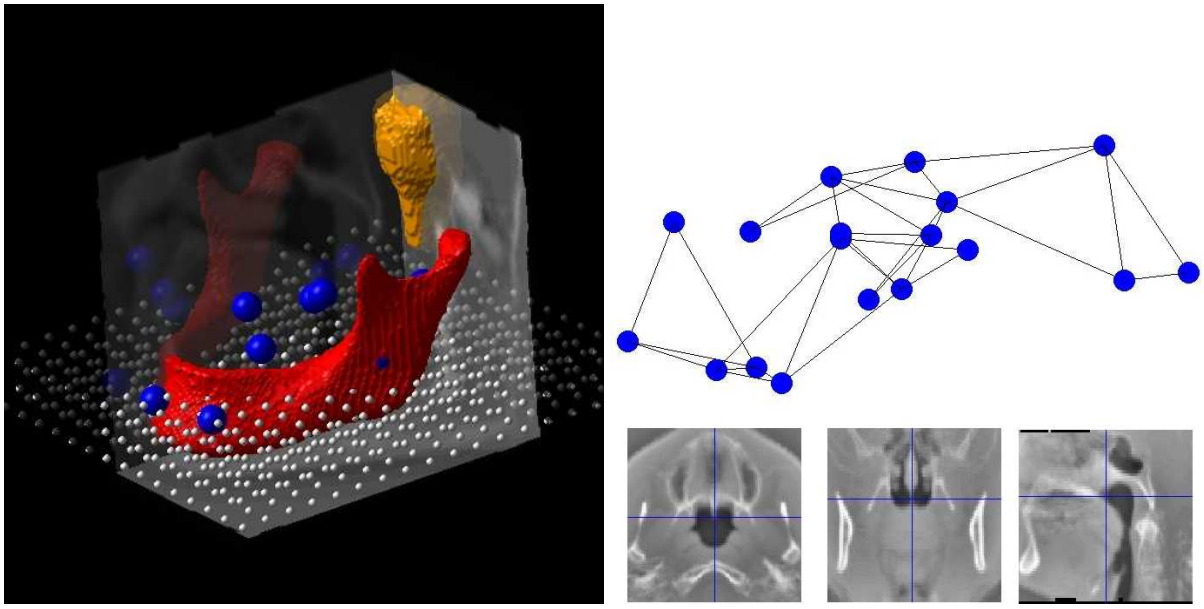

Figure 2: The figure on the left shows a subset of the control points embedded in the volume of the mean image. The larger points are those used in the parts-and-geometry model. The configuration of the pairwise relationships between the nodes is shown on the right as well profiles through one of the node centres.

2 Method

2.1 Establishing correspondence

Establishing correspondence is an important aspect of our approach. It is a prerequisite to constructing AAMs and the local regressors. We also use the resulting deformation fields in the automatic construction of the parts-and-geometry model.

We use the groupwise method of Cootes *et al.*[4] to perform non-rigid registration giving a dense set of control points which are located in corresponding locations within each image in the set being registered (see Figure 2). As in [1] we use the supplied labelled data to create two plane images in which the first plane is the binary label, and the second is a locally normalised gray level image.

The next section describes how a subset of these control points are used in the construction of the parts-and-geometry model, and sections 2.3 and 2.5 describe how they are are used in the construction of AAMs and local regressors. An important feature of our method is that using these control points to construct both the AAM and the parts-and-geometry model facilitates the use of the parts-and-geometry model in the initialisation of the AAM. This is described in section 2.4 .

2.2 Parts-and-geometry model

Parts-and-geometry models are commonly used in computer vision e.g. [6]. The parts are feature detectors centred on a particular voxel and the geometry is defined by pairwise relationships between the positions of the features (see Figure 2). In this section we describe the selection of feature points, construction of the parts-and-geometry model, and how to match it to an image.

Feature point selection

The correspondences obtained from the registration allow us to define a reference space (the mean of the control points of the training images). A reference image is constructed by warping each training image into the reference space and computing the average. The dense deformation fields define a mapping from the reference to each training image. This allows us to compute the point in each training image corresponding to a given point in the reference image.

The AAM includes a shape model and in our implementation this is obtained from the positions of the control points. If we can locate a sufficiently large subset of the model nodes (control points) well, we can fit the shape model to them to initialize the AAM search. Thus we aim to select such a subset automatically. The choice of subset will depend on the method used to locate individual nodes. Though more sophisticated local feature models could be used, in the following we use simple normalized correlation with a suitably sized template.

To select the best nodes and template sizes we consider a range of templates of different sizes taken from each node in the reference image. We use each to search the training images and select those which are more reliable at locating the equivalent point in each training image.

For a particular node we can construct a feature detector based on a region of size $(2L_x+1)$ x $(2L_y+1)$ x $(2L_z+1)$ centred on the node. For each training image I_i, there exists a deformation field Θ_i allowing a mapping of space between it and the reference. We apply the feature detector to I_i to obtain a response image R_i. The local peaks of the response image are located and ranked according to the strength of their response. Let $\mathbf{p}_k$ specify the position of a node in the reference. Its position in I_i and R_i is $\Theta_i(\mathbf{p}_k)$. We can then define a function D that computes the distance between a local peak located at $\mathbf{p}_l$ in the space of R_i and the expected position of a node in this image:

$$D_i(\mathbf{p}_k, \mathbf{p}_l) = ||\Theta_i(\mathbf{p}_k) - \mathbf{p}_l|| \tag{1}$$

For a good detector the best response will be close to the true position $\Theta_i(\mathbf{p}_k)$ in every image in the training set.

Local feature detectors are built centred on every node of the shape model mesh at a range of sizes and at different resolutions of the reference image. Each is then evaluated on a set of images. The success rate in locating the nodes and the average value of D_i over the set are computed and used to evaluate the reliability of each node. The N most reliably located nodes can be selected to be used in AAM initialisation. An algorithmic description of the process followed is given below:

For each node, at each region size and image resolution:

1. Build a feature detector for the given node
2. For each image:
   (a) Build an image pyramid and select appropriate level
   (b) Apply the feature detector to obtain a response image
   (c) Locate local peaks and rank by their response
   (d) Compute the distance of peaks to the ideal position using equation 1
   (e) Record the rank of the peak with smallest D_i.

The feature detectors are then ranked by the average rank of the best match (computed in step ‘e’ above), then by the average value of the D_i (from step ‘d’). On application to the training data a number of detectors

were found to always have their best response closest to the true position. These were then ranked by the average positional error when selecting a subset of good detectors. We also desire that the detectors are spread around and not clustered around a region. We address this by selecting the best detector then iteratively selecting the next best detector not withing a radius r of the current set of chosen detectors.

Construction
We automatically construct the parts+geometry model from a set of local feature models as follows. Using the reference image and the selected feature points in the reference frame we automatically define a set of connecting arcs based on the distances between the nodes. We use a variant of Prim's algorithm to obtain the minimum spanning tree, where each node has two parent nodes, rather than one. This involves creating the first arc from the two nodes which are closest together. We then repeat the following steps until all nodes are linked:

- compute the sum of the distances of each unlinked node to the closest two nodes in the current linked set
- select the node which has the minimum such distance, and link it to the two closest nodes in the linked set

This leads to a topology which allows a variant of the dynamic programming algorithm to efficiently find the global maxima of the cost function in Equation (2).

Matching to an image
Let $\mathbf{p}_i = (x,y,z)$ be the proposed position of a patch. Let $p_i(I|\mathbf{p}_i)$ be the probability that patch i matches to image I at the given location. Let $p_{ij}(\mathbf{p}_i,\mathbf{p}_j)$ be the probability that two patches i and j have the given positions. Assume that we have modelled this pairwise relationship for each pair $(i,j) \in A$, where A is a set defining the arcs in a graph representing the model.

To match such a model to an image, we search the image for candidate positions for each patch, then select one for each patch so as to optimise

$$C = \sum_{i=1}^{k} \log p_i(I|\mathbf{p}_i) + \sum_{(i,j)\in A} \log p_{ij}(\mathbf{p}_i,\mathbf{p}_j) \tag{2}$$

A range of discrete graph based solvers are available to find the optima for such a cost function, their efficiency depending on the complexity of the topology of the connections in the graph, A. For simple topologies there are fast, guaranteed optimal solutions. For instance, if each part is connected to one parent (but may have many child connections), we have a tree structure, and a variant of dynamic programming can be used to quickly find the optima in time of order $O(NM^2)$ (for M candidates for each of N nodes).

In the following we use a more complex variant, similar to that used by [7], in which a network is created where each node can be thought of as having at most two parents. The optimal solution for this can be obtained with a variant of dynamic programming, in $O(NM^3)$ time. If M is modest, this is still fast.

2.3 Active appearance models

An Active Appearance Model is a statistical model of both the shape of a structure and its appearance, together with an algorithm for matching it to an image. The model is capable of synthesising an image of the object of interest, and the residual differences between the synthesised image and the target image are used to drive the search.

We construct the shape model by aligning the sets of control points on each image and applying Principal Component Analysis (PCA) [3]. A statistical model of texture is constructed by warping each grey-level image into the reference space and appling PCA to the resulting textures. An appearance model [2] is a combination of the shape and texture models, with the form

$$\begin{aligned} \mathbf{x} &= \bar{\mathbf{x}} + \mathbf{Q}_s \mathbf{c} \\ \mathbf{g} &= \bar{\mathbf{g}} + \mathbf{Q}_g \mathbf{c} \end{aligned} \tag{3}$$

Where $\bar{\mathbf{x}}$, $\bar{\mathbf{g}}$ are vectors of the mean shape and mean texture, $\mathbf{x}$, $\mathbf{g}$ are the shape and texture vectors in the reference frame, $\mathbf{Q}_s$, $\mathbf{Q}_g$ are matrices describing the modes of variation derived from the training set, and $\mathbf{c}$ is a vector of parameters controlling both shape and texture.

Appearance models can be matched to new images rapidly using the Active Appearance Model algorithm [2]. This seeks to minimise a sum-of-squares problem of the form

$$F(\mathbf{p}) = |\mathbf{r}(\mathbf{p})|^2 = \mathbf{r}^T \mathbf{r} \tag{4}$$

where $\mathbf{p}$ contains the t model parameters and $\mathbf{r} = \mathbf{r}(\mathbf{p})$ is a function returning the n_g residual differences between model and data for parameters $\mathbf{p}$. By making assumptions about the Jacobian, a fast updating algorithm can be derived which can match the model to a new image in a few iterations.

2.4 Initialising the AAM

The positions of the nodes (image patches) of the parts-and-geometry model are a subset of the points defining the shape part of the AAM. Therefore, given a set of positions for each node on the parts-and-geometry model, the control points of the AAM can be fitted to them by finding the pose and shape parameters which minimise the distance between the equivalent points of the shape model and the parts-and-geometry model. Once initialised, the usual AAM algorithm can be applied to best match the whole model to the image data.

We found that the location of points by the parts-and-geometry model was very good – 98.5% success rate in leave-one-out experiments using the training data (as opposed to 82.9% without the geometric constraints). However, we built in added robustness by obtaining the Euclidean distance between the points of the parts-and-geometry model and the equivalent points of the shape model after the fit. The point with the largest error is discarded and the model re-fitted to the remaining points. This is repeated until the mean error is below a given threshold or a particular number of points have been discarded.

2.5 Local linear regressors

The result of the AAM search is an approximation of the query image. For our particular case we are interested in the shape part which gives a correspondence between the reference space and the query image. It allows a probability image of the object of interest computed in the reference space to be warped into the query image. Thresholding this at 0.5 gives a binary segmentation.

However, for a variety of reasons such as poor initialisation, un-modelled image structures or a limited number of examples in the training set the AAM result can be suboptimal and the use of local regressors [1] is an attempt to address this.

Local regressors allow the generation of a mean probability image based on pixel intensities warped into the reference frame. The intensities of the training images are normalised and warped into the reference frame

using the correspondences obtained from the groupwise registration. For any given voxel in this frame we then have a set of probabilities, p_i, $i = 1..n_{images}$, that it is inside the object (by warping the binary label images) together with corresponding vectors of intensity values $\mathbf{g}_i$ sampled in the region around the voxel.

We then perform linear regression to learn a function to estimate the probability given the intensity pattern

$$p = f(\mathbf{g}) = \mathbf{a}^T\mathbf{g} + d \tag{5}$$

This is repeated for every voxel near the boundary. Voxels away from the boundary are assumed to have either p=0 (outside) or p=1 (inside).

2.6 The full segmentation pipeline

Given a query image, we perform the following:

- Identify the patch locations using the parts-and-geometry model
- Initialise a global AAM encompassing both mandible and brainstem
- Match the global AAM to the query image to estimate correspondence between the image and model
- For each local model:
  - Use the global estimate of correspondence to initialise the local model in the query image
  - Match the local model to refine the estimate of correspondence
  - Use the refined estimate to warp a normalised version of the query image into the model frame
  - Use the voxel probability estimators (Eq.5) to compute a probability image in the model frame
  - Use the correspondences to warp this probability image back into the query image frame
  - Obtain a labelled image by thresholding the result at ≥ 0.5

3 Experiments

Firstly, in an attempt to improve statistical power we reflected the 10 supplied training images about the $y-$axis to double the training set to 20. All images were registered as described in section 2.1. Three sets of registrations were carried out. In the first the mandible and brainstem labels were combined and the image region encompassed by them (as well as a 10 voxel border region) were registered. This process was repeated independently for the mandible then the brainstem. The global AAM was constructed using the correspondence obtained from the registration of the two structures combined and the local models and regressors from the registrations of the individual structures.

The patches for the parts-and-geometry model were obtained from the subimages containing both the mandible and brainstem, and 18 patches were found to be located with high reliability across the training set. We performed segmentation on the test data using the above components as described in section 2.6.

Dataset No.	Mean HD	Median HD	No. of slices (HD $>$ 3 mm)
11	16.16	8.56	39 (39)
12	21.58	10.20	40 (39)
13	86.09	91.60	35 (35)
14	22.01	13.16	34 (34)
15	19.18	8.96	37 (37)
16	13.30	5.94	35 (35)
17	10.08	4.88	43 (39)
18	21.52	13.82	37 (36)

Table 1: Hausdorff distance (HD) statistics for mandible segmentation in the testing datasets.

Dataset No.	Average slice OV	Median slice OV	Total volume OV
11	63.5 %	68.6 %	68.1 %
12	67.2 %	69.1 %	74.3 %
13	21.2 %	15.1 %	22.0 %
14	65.0 %	66.6 %	78.3 %
15	65.6 %	70.3 %	72.9 %
16	73.4 %	79.8 %	81.2 %
17	79.8 %	82.9 %	82.9 %
18	68.9 %	75.4 %	75.1 %
Mean excluding Subject 13			76.1% $\pm$ 5.1

Table 2: Overlap (OV) statistics for mandible segmentation in testing datasets.

4 Results

The results below were obtained after submitting our results to the organisers. The only amendments are that mean and standard deviations have been added to the last columns of Tables 2 and 4. Tables 1 and 2 show the Hausdorff distance and Dice overlap values obtained when the result of our method for the mandible was compared with a manually defined gold standard. Tables 3 and 4 are equivalent values for the brainstem.

Dataset No.	Mean HD	Median HD	No. of slices (HD $>$ 3 mm)
11	5.12	4.88	28 (27)
12	5.10	4.88	29 (27)
13	-	-	-
14	9.54	9.11	30 (29)
15	29.72	32.79	18 (18)
16	3.78	3.52	27 (16)
17	4.73	3.91	27 (18)
18	4.26	4.17	29 (22)

Table 3: Hausdorff distance (HD) statistics for brainstem segmentation in the testing datasets.

Dataset No.	Average slice OV	Median slice OV	Total volume OV
11	78.2 %	80.1 %	79.4 %
12	84.7 %	84.6 %	86.2 %
13	-	-	-
14	70.1 %	75.4 %	70.6 %
15	21.1 %	25.6 %	18.6 %
16	84.8 %	86.5 %	87.6 %
17	87.1 %	88.2 %	86.7 %
18	82.4 %	81.9 %	80.9 %
Mean excluding Subject 13			72.9% $\pm$ 24.6

Table 4: Overlap (OV) statistics for brainstem segmentation in testing datasets.

5 Discussion

This method based on a framework shown to give good performance on subcortical structures in the brain [1]. However, the results obtained here are not as good as those for the subcortical structures. As the results obtained by other participants in the challenge were not available at the time of writing, we cannot determine the role the quality of the dataset played in the level of performance. However, in a recent publication Han *et al.*[8] report median Dice overlaps of about 90% for the mandible and 80% for the brainstem.

Our method failed to segment both structures in Subject 13 and the brainstem in subject 15. This was because it assumed that both these structures would always be present in the images to be segmented, which was not the case. The small number of supplied images is a significant contributor to the performance as it violated the assumption that the training set is representative of the class of structures being modelled. Other reasons for suboptimal performance include artefacts in the images due to fillings and missing teeth in some subjects. The average running time is 17 minutes on a 2GHZ pentium 4 system with 2GB RAM. The main contributor to this is the size of the images (about $512 \times 512 \times 190$ voxels)

The advantages of this method are that it is general and portable in that the training data is encoded in the model so images of specific patients are not needed once the model is built. Increasing the amount

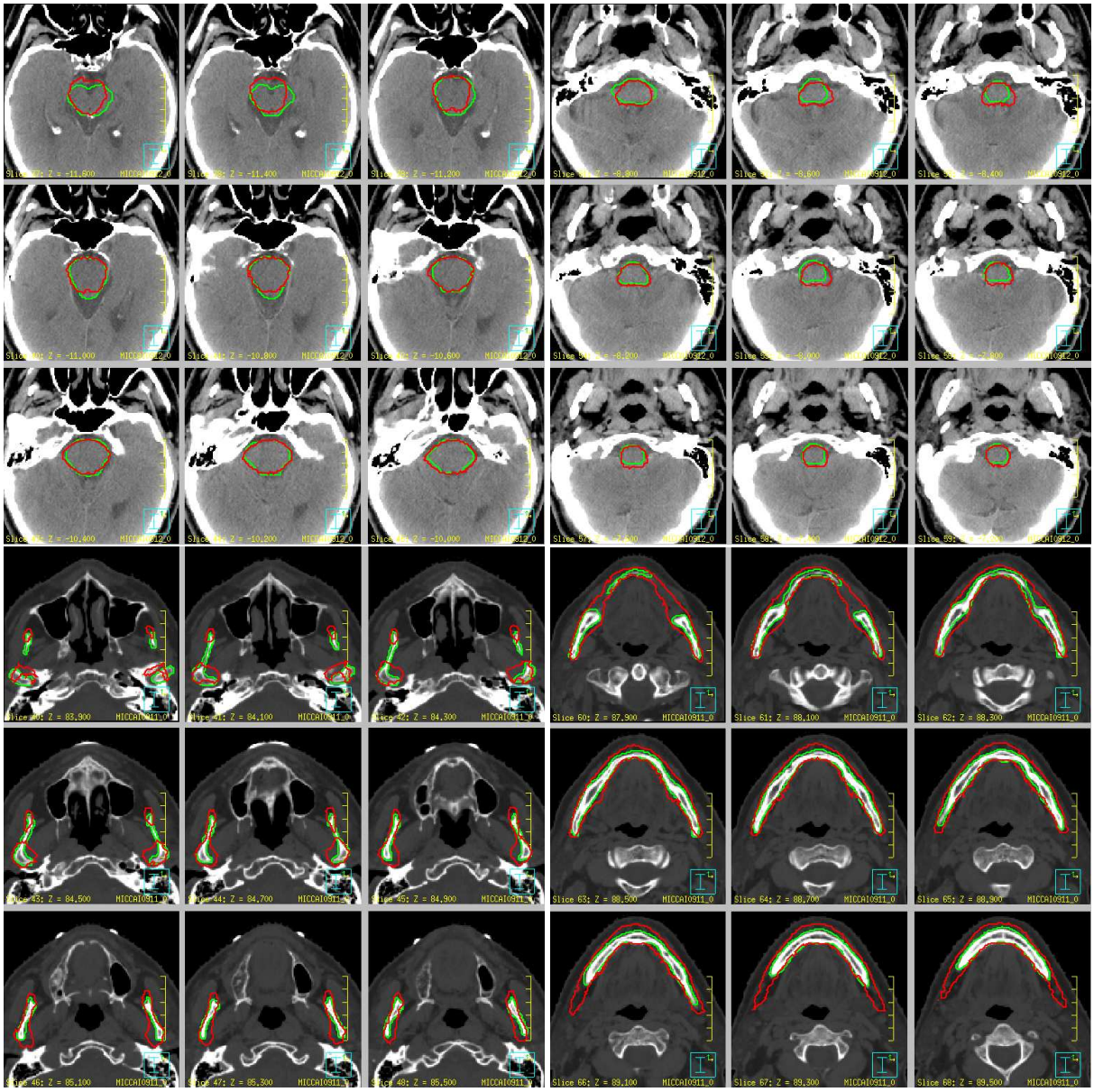

Figure 3: Official results showing the result of our algorithm (red) and that of the manually segmented gold standard for Subject 12.

of training data will improve the models, but will not result in significant increases in running time. The statistical model of shape incorporates prior knowledge and reduces sensitivity to image artifacts and missing teeth. Furthermore, user interaction can be easily incorporated using the shape part of the AAM. If speed is an issue, cropping the image to the region containing the mandible and brainstem after global search, and performing the local searches and regression in parallel can reduce the running time substantially.

However, the current implementation suffers from the fact that we assume the entirety of the mandible and brainstem are present in all images to be segmented. To account for images without the full field of view the nodes used to build the parts-and-geometry model can be restricted to a subregion that will be present in all images or the formulation of the solver can be changed to allow for missing data. However, building such robustness into the shape models is not straightforward. A possible approach is to use the model to predict the missing data.

In conclusion, our method gives encouraging results on this data set and we believe that with minor modifications and a larger training set the method could be made to work substantially better.

References

[1] K O Babalola, T F Cootes, C J Twining, V Petrovic, and C J Taylor. 3D brain segmentation using active appearance models and local regressors. In *Proceedings of MICCAI*, volume 5241 of *LNCS*, pages 401–408, 2008. 1, 1, 2.1, 2.5, 5

[2] T F Cootes, G J Edwards, and C J Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001. 1, 1, 2.3, 2.3

[3] T F Cootes, C J Taylor, D H Cooper, and J Graham. Active Shape Models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, January 1995. 2.3

[4] T F Cootes, C J Twining, V Petrovic, R Schestowitz, and C J Taylor. Groupwise construction of appearance models using piece-wise affine deformations. In *Proceedings of 16th British Machine Vision Conference, Oxford*, pages 879–888, 2005. 2.1

[5] D Cristinacce, T Cootes, and I Scott. A multi-stage approach to facial feature detection. In 15th *British Machine Vision Conference, London, England*, pages 277–286, 2004. 1

[6] P F Felzenszwalb and D P Huttenlocher. Pictorial structures for object recognition. *Int. Journal of Computer Vision*, 61(1):55–79, 2005. 2.2

[7] P F Felzenszwalb and D P Huttenlocher. Representation and detection of deformable shapes. *IEEE PAMI*, 27(2):208–220, 2005. 2.2

[8] X Han, M S Hoogeman, P C Levendag, L S Hibbard, D N Teguh, P Voet, A C Cowen, and T K Wolf. Atlas-based auto-segmentation of head and neck CT images. In *Proceedings MICCAI*, volume 5242 of *LNCS*, pages 434–441, 2008. 5

[9] S C Mitchell, J G Bosch, P F Boudewijn, B P F Lelieveldt, R J van der Geest, J H C Reiber, and M Sonka. 3-D active appearance models: Segmentation of cardiac MR and ultrasound images. *IEEE Transactions on Medical Imaging*, 21(9):1167–1178, 2002. 1

[10] H H Thodberg and A Rosholm. Application of the Active Shape Model in a commercial medical device for bone densitometry. In T F Cootes and C J Taylor, editors, *Proceedings of the 12th Britich Machine Vision Conference*, volume 1, pages 43–52, September 2001. 1