Graphical Shape Templates for Automatic Anatomy Detection with Applications to MRI Brain Scans

Yali Amit

Abstract— **A new method of model registration is proposed using graphical templates. A decomposable graph of landmarks is chosen in the template image. All possible candidates for these landmarks are found in the data image using robust relational local operators. A dynamic programming algorithm on the template graph finds the optimal match to a subset of the candidate points in *polynomial time*. This combination—local operators to describe points of interest/landmarks and a graph to describe their geometric arrangement in the plane—yields fast and precise matches of the model to the data with no initialization required. In addition, it provides a generic tool box for modeling shape in a variety of applications. This methodology is applied in the context of T2-weighted magnetic resonance (MR) axial and sagittal images of the brain to identify specific anatomies.**

Index Terms—**Graph matching, model registration, shape and object representation.**

I. Introduction

When observing a medical image it is relatively easy for the human to identify the important anatomies and their subcomponents. For example, in the axial MRI brain scan in Fig. 3, one immediately identifies not only the ventricles as a whole but the different subanatomies such as the frontal and posterior horns. This process of identification can be called structural recognition or model registration; together with the recognition of the object comes an identification of its components and their arrangement in the image.

The local topography at the tip of the right frontal horn of the lateral ventricle has the form of an "outlet" pointing approximately northeast, as indicated by the level curves of the image around that point [see Fig. 1(a)].

There are many other locations in the image which have a very similar local topography, for example, some of the sulci [see Fig. 1(b)]. Thus, the tip of the right frontal horn of the lateral ventricle can not be identified only through these local characteristics. The position relative to a number of other topographies, such as that of the tip of the other frontal horn and the two posterior horns is crucial for this identification. In this paper we attempt to formalize these ideas and describe how they can be incorporated in an algorithm for automatic model registration and anatomy identification.

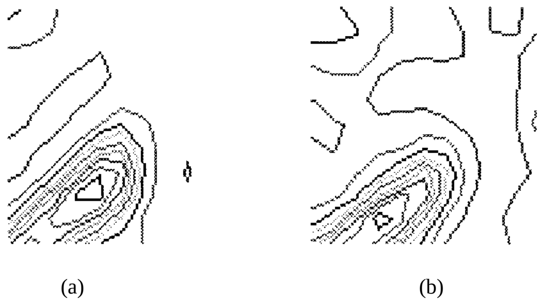

Fig. 1. (a) Contour plot of neighborhood of frontal horn. (b) Contour plot of neighborhood of a sulcus.

The problem of model registration in medical imaging is becoming of growing interest. One of the most noted examples in the context of brain imaging is the human brain atlas of [1], where several points identified by the user, serve to generate a unified coordinate system for the human brain. Registration provides an automatic means of identifying the various components of the object in the image, and in some cases of segmenting the image. This is extremely important in the context of data fusion, e.g., positron emission tomography (PET) scans with magnetic resonance imaging (MRI) scans, where metabolic information needs to be integrated with anatomical information. Current research into brain function, whether using functional PET or functional MRI, needs to compare activity regions across patients. All these studies require some form of registration. Model registration also provides a means of studying the variability in the family, classifying subgroups and identifying abnormalities.

Thus far, model registration has primarily been achieved through elastic image matching or image warping in two and three dimensions. See for example [2] for hand X rays and [3]–[5] for MRI images of the brain. A comparison of the different approaches can be found in [6]. Currently three-dimensional versions of these algorithms are being used for brain mapping. One-dimensional (1-D) elastic models have also been extensively studied; see [7]–[9].

There are several limitations to these elastic models. First the matching criterion seeks to minimize the mean square difference between the pixel intensities of the deformed template and that of the data image. On one hand, this is a very well defined criterion, however, it does not ensure that specific points of interest or landmarks be matched with great precision. This later criterion is relatively clear to the human eye but very hard to formulate in mathematical terms. Second,

Manuscript received November 7, 1995; revised October 24, 1996. This work was supported in part by the Army Research Office (ARO) under Grant DAAL03-92-G-0322. The Associate Editor responsible for coordinating the review of this paper and recommending its publication was V. Johnson.

The author is with the Department of Statistics, University of Chicago, 5734 University Avenue, Chicago, IL 60637 USA (e-mail: amit@galton.uchicago.edu).

Publisher Item Identifier S 0278-0062(97)00979-8.

the deformations being used are generic or nonparametric in nature and do not depend on the specific family of objects. Third, because of the inherent nonlinearity of the problem, and the fact that the deformations are high dimensional, the computational tools for calculating the match must use relaxation techniques which run the risk of converging to a local minimum which corresponds to a poor match. This is a serious problem in the 1-D elastic algorithms which fail entirely without explicit user initialization.

The main premise in this paper is that structural recognition or model registration can be achieved through graphical models describing the *global planar arrangement of a small number of landmarks*. Each landmark is defined by some local operator which loosely describes the topography of the image intensity surface in a small neighborhood. We will be exploiting the fact that there is a very small chance of finding a specific geometric arrangement of landmarks associated with specific operators, in an arbitrary image or an arbitrary location in an image. The only probable occurrence of such an arrangement is at the specific structure from which it is derived.

All pixels that are activated by a local operator are candidates for the corresponding landmark. It is the global planar arrangement of the landmarks which singles the true positives from the many false positives which are activated by the operators. This is achieved computationally by creating a decomposable graph of triangles, which partially describes the global arrangement of the landmarks, and then using dynamic programming to find the optimal match to the large collection of landmark candidates, in polynomial time.

Dynamic programming has been used in medical image analysis in applications such as artery tracking, see for example [10]. These settings are 1-D in nature, and the constraints enforced by the underlying graph are all local. To our knowledge, this is the first attempt to incorporate this tool in an inherently two-dimensional and nonlocal imaging problem.

In principle, the description of the global arrangement is translation, scale, and rotation invariant. However, the local operators themselves are not scale and rotation invariant, although they are robust to small rotations and a wide range of scale. Hence, the general orientation and approximate scale of the object are assumed to be known, as is typically the case in medical imaging. Apart from orientation and scale no initialization is required.

Once the graph has been matched to the data it is possible to create a *planar* map through interpolation as in [11]. Furthermore, this interpolated map can serve as an initial point for the image warping algorithms where the correct match of the landmarks is ensured. Alternatively, the matched landmarks can serve as an initialization to a variety of 1-D elastic algorithms. In another application, graphical models describing geometric arrangements of landmarks, are used in randomized decision trees for shape recognition; see [12].

In [13] the graphical template was first introduced and applied to hand X rays. In this paper some further aspects of the implementation of this method are investigated including new forms of local operators, different ways to choose the graph, new forms of cost functions, and ways to improve the robustness of the method and reject false matches. The details are provided in Section II together with experimental results on MRI brain scans. One immediate payoff of this method is that the landmarks defining the standardized coordinate system of the brain as suggested in [1] can be automatically identified.

Modeling variability in terms of the relative locations of the landmarks yields a more specific and lower-dimensional description of the variability within a certain anatomy. Ultimately, this opens the possibility of truly estimating the parameters of a credible prior distribution. With this goal in mind, a Bayesian formulation of the method is presented in Section III.

In this context we mention work by Wilson [14], who defines multiscale hierarchical priors, describing spatial configurations of various types of local features at different scales. The local features are defined through differential operators on Gaussian scale-space; the configurations are defined through functions defined in terms of scale-space distance between pairs of features. These models involve large numbers of feature points and complex graphs and can be viewed as providing more realistic priors. The payoff is that optimization or simulation again involve intensive relaxation methods with problems similar to those mentioned above. However, it is possible to implement the graphical template algorithm described here using the local features proposed in [14]. Indeed, it could then be viewed as an efficient “pruning” both of the modeling and of the optimization problems posed by the full Bayesian model.

The method described below can be thought of as a generic *tool box* for generating shape models of anatomies. Ultimately, for each image family a graph model will be matched to any incoming data image, yielding an automatic identification of the various components.

II. The Graph-Matching Algorithm

A. The Local Operators

A collection of landmarks is chosen in the template image (see Fig. 3) and a local operator is chosen for each landmark. The local operators are designed to be robust, and hence, crude; descriptions of the local topography of the pixel intensity function in the neighborhood of the landmark.

A certain size neighborhood is chosen, say $m \times n$, together with an array L of 1’s, -1’s, and 0’s of those dimensions. The *sign* of the difference between the intensity at a given pixel i, j, and the intensity at each of the pixels in its $m \times n$ neighborhood is calculated, to yield an array $A^{(i,j)}$ of 1’s and -1’s, at pixel i, j. If the percentage of matched 1’s and -1’s between L and $A^{(i,j)}$ is above a prescribed threshold the pixel is considered a candidate for the corresponding landmark. The 0’s region in L represents a region where the results of the $A^{(i,j)}$ are ignored. In Fig. 2 are the 16 masks used in creating the models. White indicates intensity greater than center value; dark, intensity less than center value; grey, ignored.

Loosely speaking, these 16 masks identify “turns” of level curves of the image (see Fig. 1) pointing in eight different orientations, and two gradient-flow directions per orientation. Not

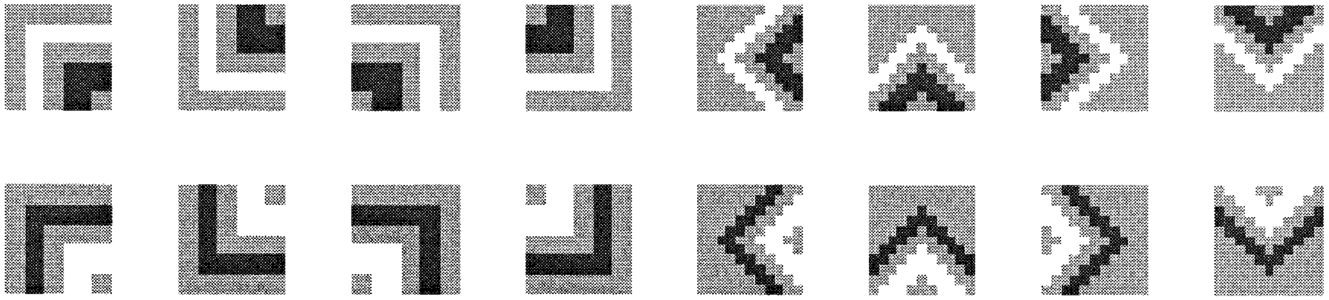

Fig. 2. Sixteen masks used for creating the graph models. White indicates intensity greater than center value; dark, intensity less than middle value; grey, ignored.

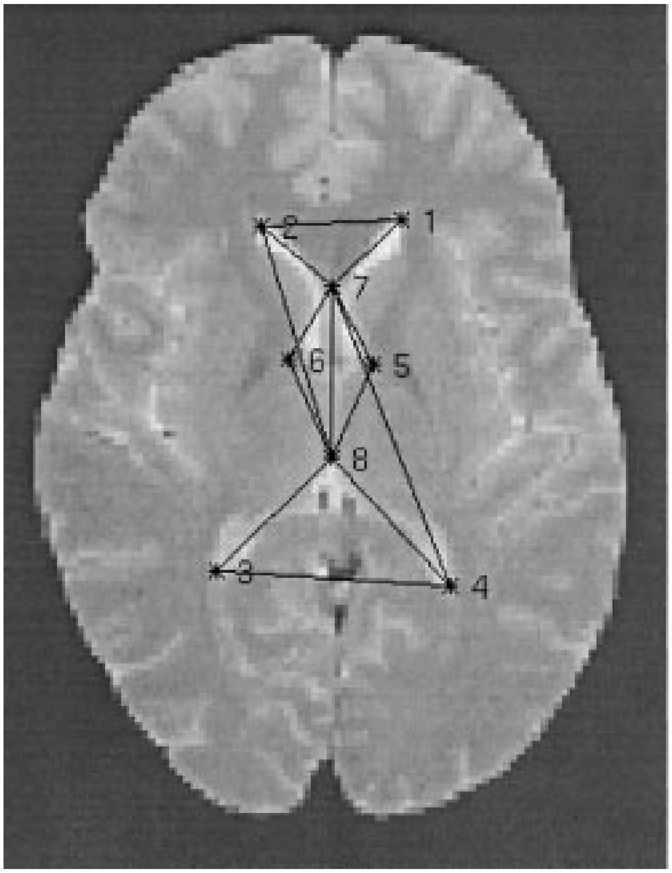

Fig. 3. The landmarks and the graph in the template image.

all masks have to be used in a given model. These operators are robust to rather significant variations in pixel intensities which are bound to occur between images as well as to smooth deformations of the local topography of the landmark neighborhood in the template image. They are also robust to a certain range of scales and small rotations. Fig. 4, bottom panel, shows 11×11 neighborhoods of the tip of the frontal horn of the lateral ventricle in four images. All four were activated by the first operator in Fig. 3. This collection can be expanded to contain other types of topographies. Other size neighborhoods can be used to get information on different scales.

The matching threshold for each mask is taken as the lowest match observed at the correct location of the landmark (pointed out by the user) over a small number of training images. The estimated minimum is used to avoid false negatives, namely, failing to find a candidate near the true location in the image for a specific landmark. This increases the number of false positives. Dealing with these is the role of the graphical model, which provides constraints on the planar arrangement of these landmarks. The more constraints are incorporated, the less likely will it be to find an erroneous configuration of landmark candidates that satisfies these constraints. Of course, if the threshold is too low, the chance of finding a candidate for any operator in any small area of the image is large enough, and the chance of finding an erroneous configuration satisfying the constraints becomes nonnegligible.

The pixels corresponding to a certain local operator typically occur in clusters. These are identified and the average location calculated. Also, to reduce the number of candidates appearing in the background, the pixel intensity at the center is required to be above some percentile of the list of pixel intensities in the image. Fig. 4(a) displays as circles the candidates found for landmark one of the model in Fig. 3 using the first mask in Fig. 4(b) and a threshold of .8. The other masks used for the landmarks of Fig. 3 are also shown in Fig. 4(b). The second mask is used for landmark two, the third mask is used for landmark seven, the fourth mask is used for landmark eight, the fifth mask is used for landmarks three and six, and the sixth mask is used for landmarks four and five.

It should be emphasized that the graph-matching algorithm can be implemented with any family of local operators.

B. Decomposable Graphs for Matching

Typically, the same local operator will be used for several landmarks, and many more candidates will be found in the data image, than landmarks associated with that operator in the model, see Fig. 4.

To find the correct match of the landmarks in the model to the candidate pixels in the image, it is necessary to introduce constraints on the relative locations of the landmarks in the plane. This can be done by defining a collection of triangles between triples of landmarks identified on the template image; see Fig. 3. A cost function is associated with each such triangle which penalizes its shape deviation from the corresponding triangle in the template image. The total cost function is the

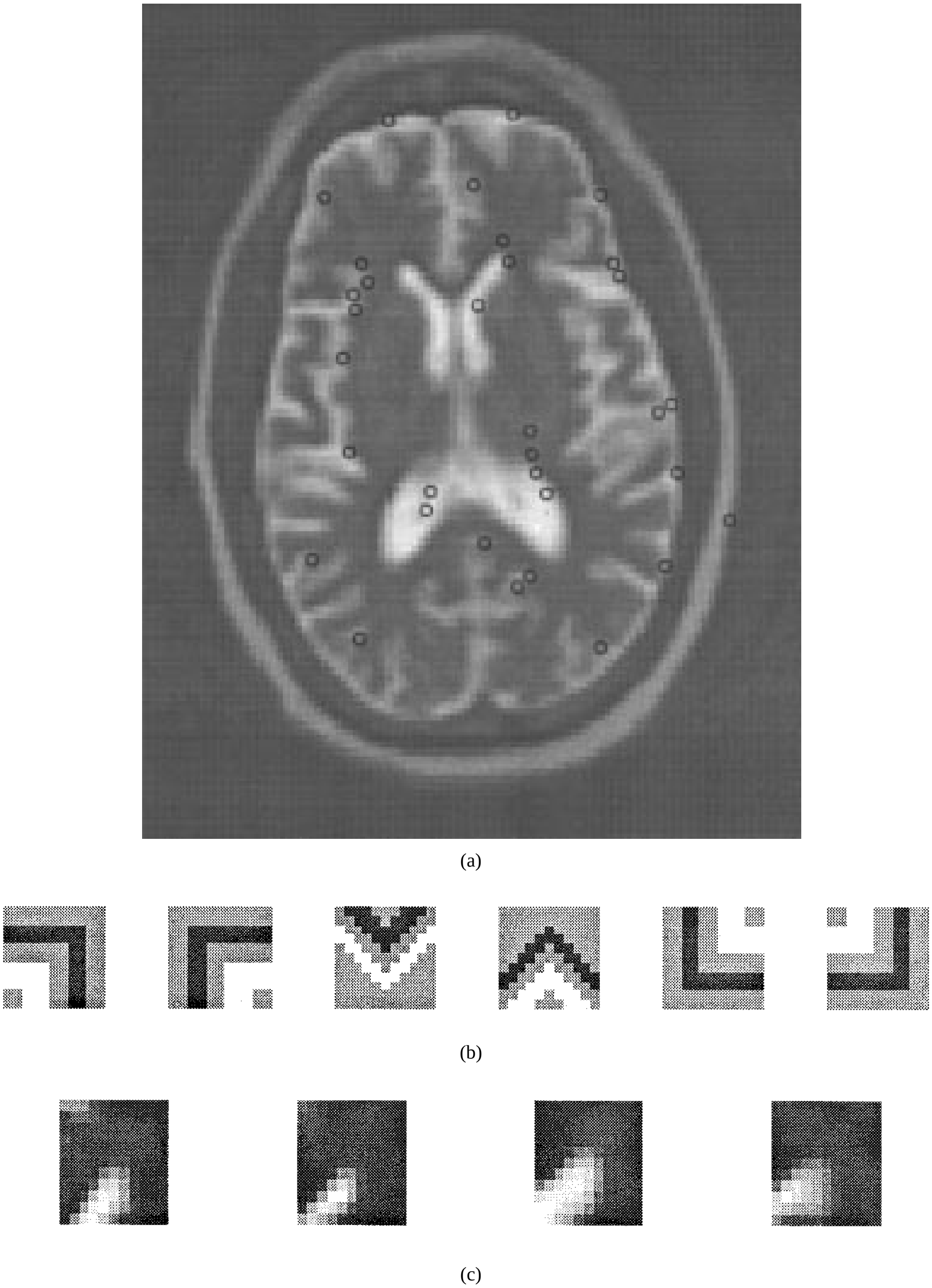

Fig. 4. (a) Candidates for the first landmark found in a data image. (b) The six masks used for identifying candidates for the model landmarks. (c) Neighborhoods of tip of upper-right ventricle in four images.

sum of the cost functions over all the triangles in the model. The collection of triangles can also be expressed as a colored graph, where the nodes are the landmarks, the color, or type of a node, is given by the type of local operator used to find its candidates. The edges in the graph exist between each two landmarks which belong to a triangle. This graph is called the template graph. In this context the deformable template is not an image but a graph model. It is important to note that these constraints need not be *local*, indeed they could involve far apart landmarks. This is in contrast to the implicit constraints used in the elasticity models which essentially penalize large changes in the local lattice elements.

C. The Cost Functions

The cost function ϕ is formulated in terms of the deviation of the three angles of the candidate triangle from the three angles of the model triangle. There are two components: hard constraints, which limit the range in which the angles can vary; soft constraints, which penalize deviations from the template angles. The hard constraints are again defined using prior information regarding the possible variations of the triangles. Soft constraints consist are the sum of the squares of the differences of the angles. Let $\theta_1, \theta_2,$ and θ_3 be the angles in the model triangle defined by the landmarks at v_i, v_j, v_k. Let η_1, η_2, η_3 be the angles in the triangle defined by $d_{\alpha_i}, d_{\alpha_j}, d_{\alpha_k}$ where d_{α_r} is a candidate for the landmark $v_r, r = i, j, k$. Then $\phi(d_{\alpha_i}, d_{\alpha_j}, d_{\alpha_k}) = \sum_{l=1}^{3}(\theta_l - \eta_l)^2$. The total cost for a candidate match $d_{\alpha_1}, \cdots, d_{\alpha_n}$ to the graph defined on the n landmarks $v_1, \cdots, v_n$ is given by

$$\Phi(d_{\alpha_1}, \cdots, d_{\alpha_n}) = \sum_{(i,j,k)\in\mathcal{C}} \phi_{ijk}(d_{\alpha_i}, d_{\alpha_j}, d_{\alpha_k}) \qquad (1)$$

where $\mathcal{C}$ is the collection of triangles in the graph.

The cost functions are translation and scale invariant. As defined above, they are also rotation invariant, however, the local operators themselves are not. Since the orientation is usually known in the present context, it is very useful to add a rotational constraint on the cost functions. For example, constraining the absolute angle of an edge to lie within a certain range of the angle of the corresponding edge in the model. Nonetheless, the matches in Fig. 6 demonstrate a robustness to small rotations which were present in the data.

D. Dynamic Programming on the Graph

Finding the optimal match then reduces to an *inexact consistent-labeling problem* [15, ch. 17], which is generically exponential in complexity. However, if the template graph is chosen so as to be decomposable, it is possible to find the optimal match in *polynomial time* using dynamic programming on the graph, see for example [16]. Decomposability in the present context means that there exists an order in which the triangles of the graph can be successively eliminated, such that each triangle in turn has a free vertex contained in no other triangle. When the free vertex and the two edges emanating from it are removed, one of the vertices of the next triangle in the order is freed, and so on.

Given a list of vertices $v_1, \cdots, v_n$ of the graph, the simplest decomposable graph—called the default graph—would have edges between $(v_i, v_{i+1}), (v_i, v_{i+2}), (v_{i+1}, v_{i+2})$ for all $i = 1, \cdots, n-2$. This is the graph corresponding to a second-order linear Markov chain. In this case, dynamic programming for optimizing an additive cost function is standard. However, sometimes other types of decomposable graphs may be useful such as the one shown in Fig. 3. The implementation of dynamic programming in this more general case is described below.

E. Example

The dynamic programming procedure to optimize the total cost function Φ as defined in (1) is illustrated using the decomposable graph of Fig. 3. For a general discussion see [13]. For each possible match $d_{\alpha_2}, d_{\alpha_7}$ for vertices v_2, v_7 find the point d_{α_1} which minimizes

$$\phi_{1,2,7}(d_{\alpha_1}, d_{\alpha_2}, d_{\alpha_7})$$

call the minimizing index $\alpha_1(\alpha_2, \alpha_7)$ and the associated cost $c_{1,2,7}(\alpha_2, \alpha_7)$. Store these two for each pair of candidates (α_2, α_7). Now for each possible match $d_{\alpha_7}, d_{\alpha_8}$ find the point d_{α_2} which minimizes

$$\hat{\phi}_{2,7,8}(d_{\alpha_2}, d_{\alpha_7}, d_{\alpha_8}) = \phi_{2,7,8}(d_{\alpha_2}, d_{\alpha_7}, d_{\alpha_8}) + c_{1,2,7}(\alpha_2, \alpha_7).$$

Note that the second term is already stored in memory. Call the minimizing index $\alpha_2(\alpha_7, \alpha_8)$ and the associated cost $c_{2,7,8}(\alpha_7, \alpha_8)$. For the triple 3, 4, 8 minimize $\phi_{3,4,8}$ to obtain an index $\alpha_3(\alpha_4, \alpha_8)$ and a cost $c_{3,4,8}(\alpha_4, \alpha_8)$ for every pair α_4, α_8. Now, for every $d_{\alpha_7}, d_{\alpha_8}$ find that $\alpha_4(\alpha_7, \alpha_8)$ which minimizes the updated local cost function on the triangle $(4, 7, 8)$ given by

$$\phi_{4,7,8}(d_{\alpha_4}, d_{\alpha_7}, d_{\alpha_8}) + c_{3,4,8}(\alpha_4, \alpha_8) + c_{2,7,8}(\alpha_7, \alpha_8).$$

Proceed through the triangles $(5, 7, 8)$ and $(6, 7, 8)$ in a similar fashion, always minimizing the updated local cost. The best $\alpha_6(\alpha_7, \alpha_8)$ is found for every pair of (α_7, α_8) with the updated cost function

$$c_{6,7,8}(\alpha_7, \alpha_8) = \phi_{6,7,8}(d_{\alpha_6}, d_{\alpha_7}, d_{\alpha_8}) + c_{5,7,8}(\alpha_7, \alpha_8).$$

The optimal pair α_7^*, α_8^* is chosen, α_6^* is taken to be the already stored index $\alpha_6(\alpha_7^*, \alpha_8^*)$, for α_5^* take $\alpha_5(\alpha_7^*, \alpha_8^*)$, for α_4^* take $\alpha_4(\alpha_7^*, \alpha_8^*)$, for α_3^* take $\alpha_3(\alpha_4^*, \alpha_8^*)$, and so on, until the entire optimal match $\alpha_1^*, \cdots, \alpha_8^*$ is recovered. This procedure will find the global minimum of (1). The memory and computation costs are of the order of nm^3 where n is the number of vertices in the graph and m is a bound on the number of candidates for each vertex.

F. Choosing the Graph

The results of matching the model in Fig. 3 are shown in Fig. 5. The matches using the default graph on a different ordering of the landmarks are shown in Fig. 6. The results are very similar.

Thus, there are many graph models which can be applied. Some graphs are less successful in conveying the appropriate geometric information on the correct arrangement of the landmarks. See Fig. 7. It is hard to precisely identify, *a priori*, which graphs will work better. It appears, however, that if two triangles are based on a common edge $\overline{ab}$, and the third points, c and c', are, broadly speaking, in the same area of the plane relative to the edge $\overline{ab}$, then the geometric information on the relative positions of c and c' can only be conveyed indirectly through the cost function ϕ on the angles of triangles abc and abc'. Since joining c and c' with an edge will violate decomposability, hard constraints are more difficult to impose. This may lead to matches in which the relative locations of c and c' do not make sense, as in Fig. 7.

G. Automatic Anatomy Identification

The location of many anatomies can now be determined relative to the matched landmarks. For example vertices one, two and three, four are at the tips of the frontal and posterior horns of the lateral ventricle. Just below landmark seven lies the septum pellucidum and just above it the trunk of the corpus callosum. The two internal cerebral veins lie near vertex eight.

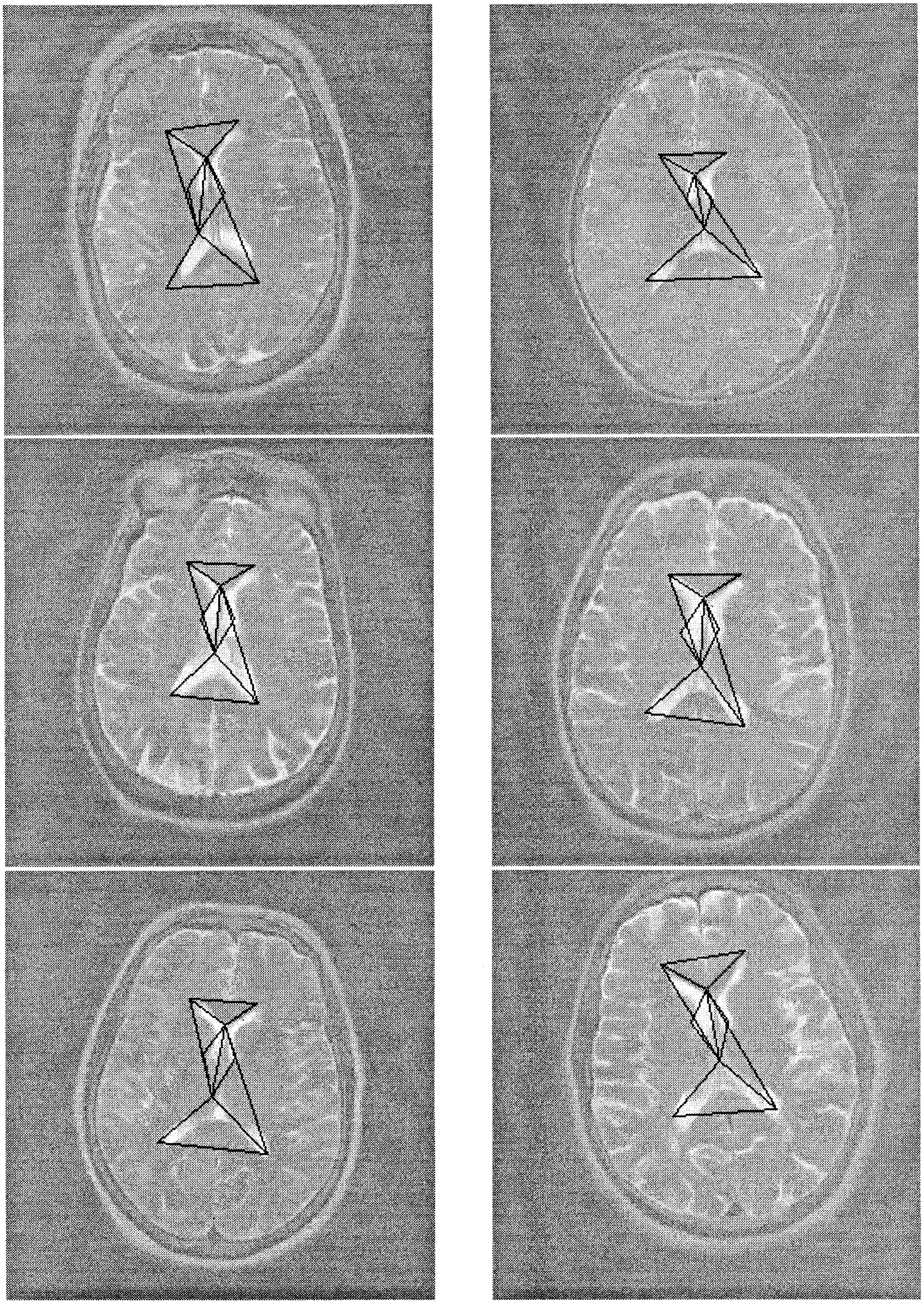

Fig. 5. The matches of the model in Fig. 3 to six axial MRI images.

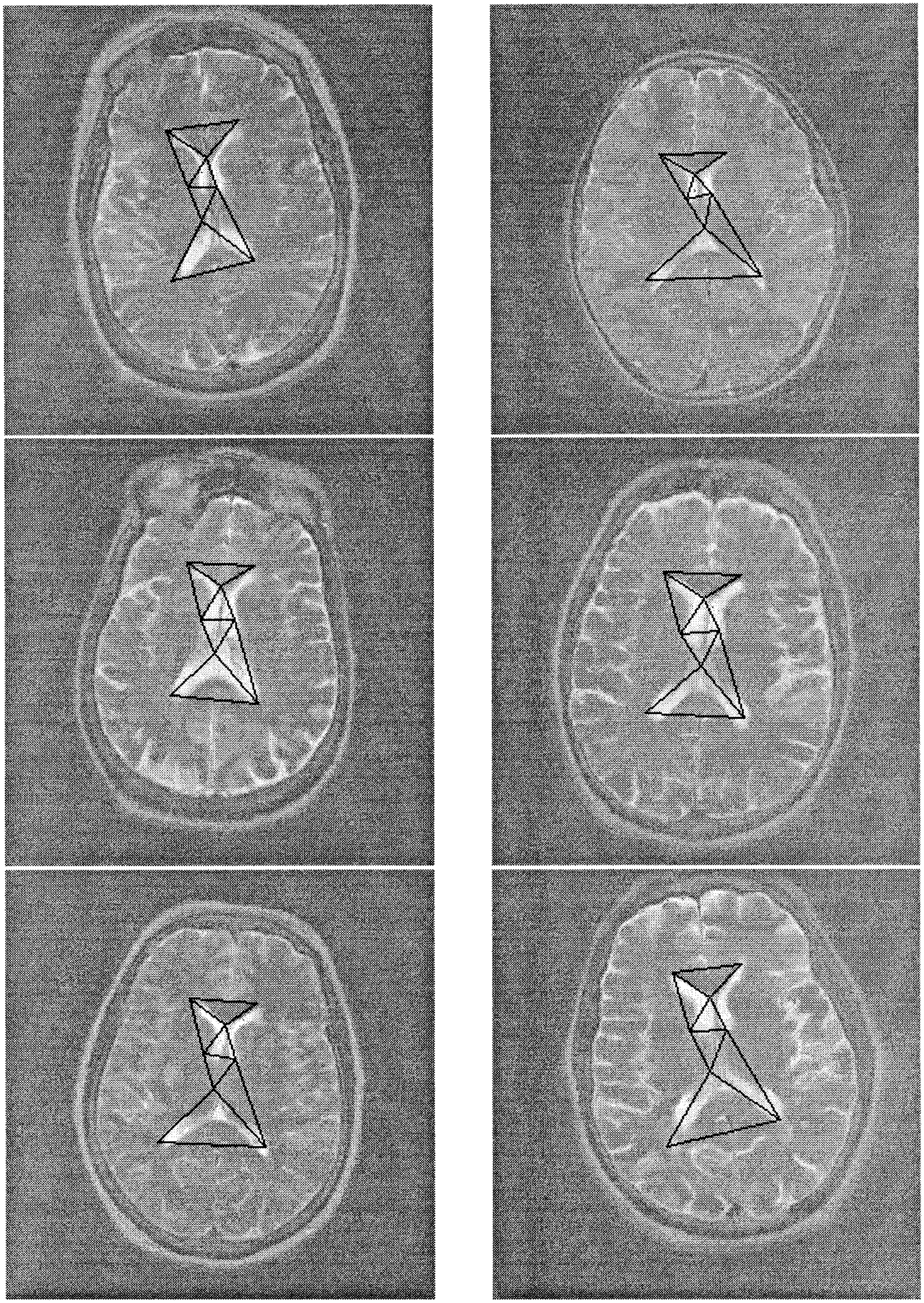

Fig. 6. Matches to six axial MRI brain images with a second-order Markov graph.

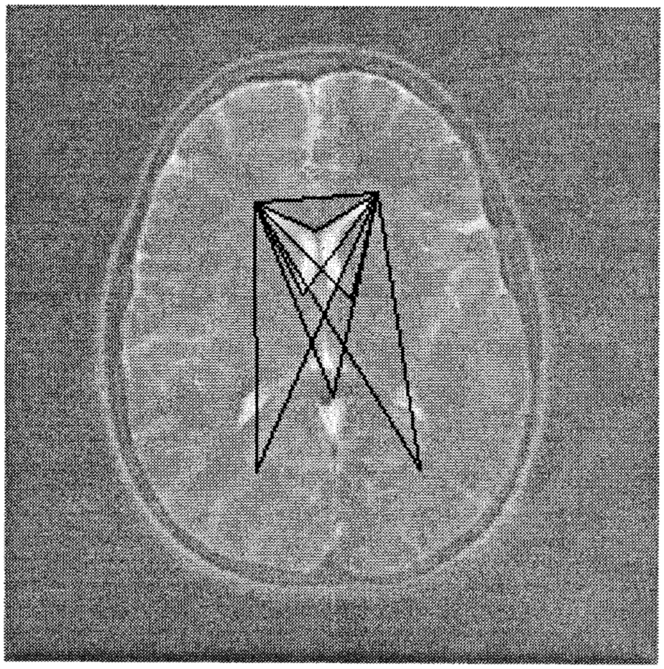

Fig. 7. An unsuccessful match with a different graph.

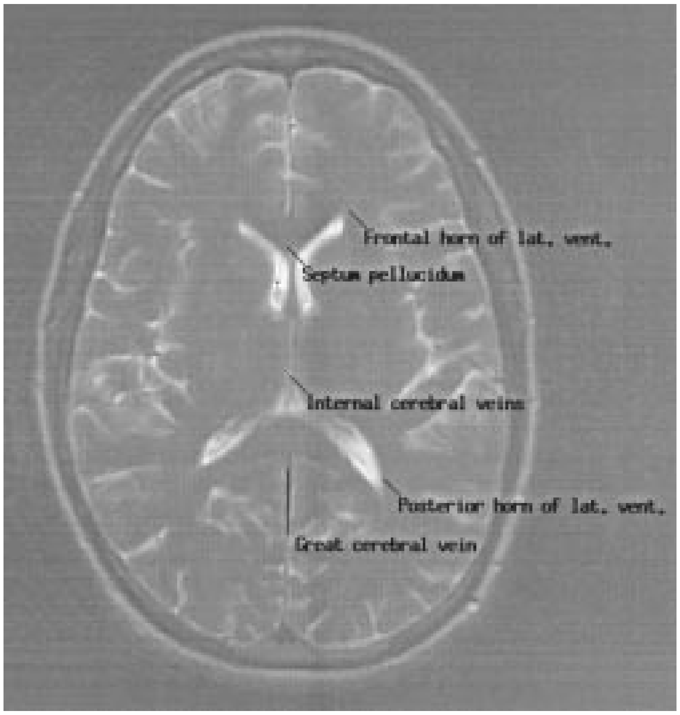

Fig. 8. Automatic identification of anatomies in an axial image.

Below vertex eight lies the splenium of the corpus callosum and near the midpoint between vertices seven and eight lies the great cerebral vein. In Fig. 8 some of these anatomies have been pointed out.

Similarly, for medial sagittal slices, Fig. 10, the nine vertices of the model are located at the fourth ventricle, the top of the pons, the frontal tip of the genu of the corpus callosum, the back of the genu, the rostrum of the corpus callosum, between the trunk and the splenium of the corpus callosum, at the bottom of the splenium, at the back of the splenium, and at the top of the cerebellum. These identifications were made using [17]. Fig. 11 shows some of these anatomies. Relative to these nine vertices, it is not hard to locate the bicommissural line of Talairach (CA-CP) and vertical line on the anterior commissure (VCA) lines which define the standard coordinate system in [1].

H. Robustness Issues

It is possible that a good match of the graph, i.e., with very low cost, will be found to erroneous candidates. The lower the threshold associated with each mask, the larger the number of candidates and the higher the chance of such an occurrence. See Fig. 9, where a graph model was used to identify the corpus callosum in sagittal MRI scans of the brain. Successful matches of this model are shown in Fig. 10.

It is useful to impose a rejection criterion for a match. This criterion involves evaluating a *new* cost function on the match proposed by the algorithm and rejecting if the cost is above a certain threshold. This cost is a sum of the original functions ϕ over *all* $\binom{n}{3}$ possible triangles between points in the model. This *global cost* function will identify erroneous matches in which the particular triangles in the model matched well but the complete configuration obtained is not similar to the one in the model. In the first two images, Fig. 9 (top row), the anatomies are easily identified by eye and yet the preferred match was not a good one. In the last four images it is not at all clear that the anatomies are present. In all six cases the global cost function was many standard deviations (SD's) away from the average, estimated over 13 correct matches, six of which are shown in Fig. 10. The average global cost over these matches was 5.1 with SD of 2.09. Of the erroneous matches the closest global cost was over four SD's above average.

It is always possible to add landmarks to the graph model. As long as the presence of these landmarks is stable throughout the family of images, this should reduce the chance of erroneous matches and hence enhance the robustness.

Another way of stabilizing the algorithm is by imposing constraints on the absolute locations of the landmark locations. This, however, eliminates scale and translation invariance and may prove sensitive to shifts such as those present in Fig. 10.

There may be ways of combining the information of several graphs, so that even if some of the vertices of each individual graph are mismatched, the combined information yields a good match.

I. Experiment

To further study the robustness of the algorithm we obtained axial T2-weighted MRI images from 68 patients of the Radiology Department at the University of Chicago Hospital. For each patient three images were provided: the eighth, ninth and tenth slices from the top. Typically one of these slices contains anatomies similar to the ones in Fig. 8. In all 204 images were obtained.

The images were divided into two groups: good images, which contained the anatomies in questions; bad images, which did not. There were 57 good images and 147 bad images. Most patients had one good image but in some cases none

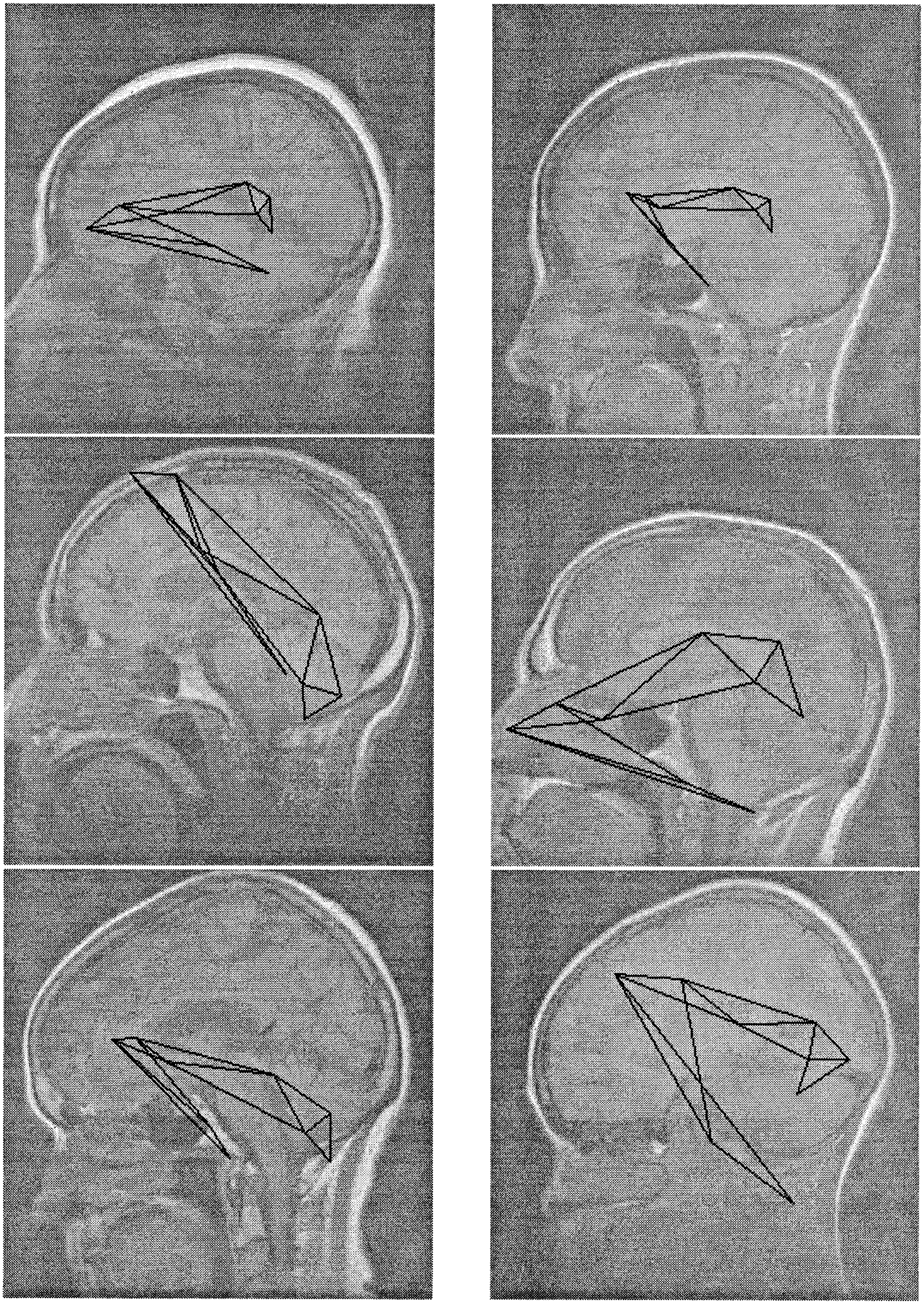

Fig. 9. Erroneous matches to six sagittal MRI brain images.

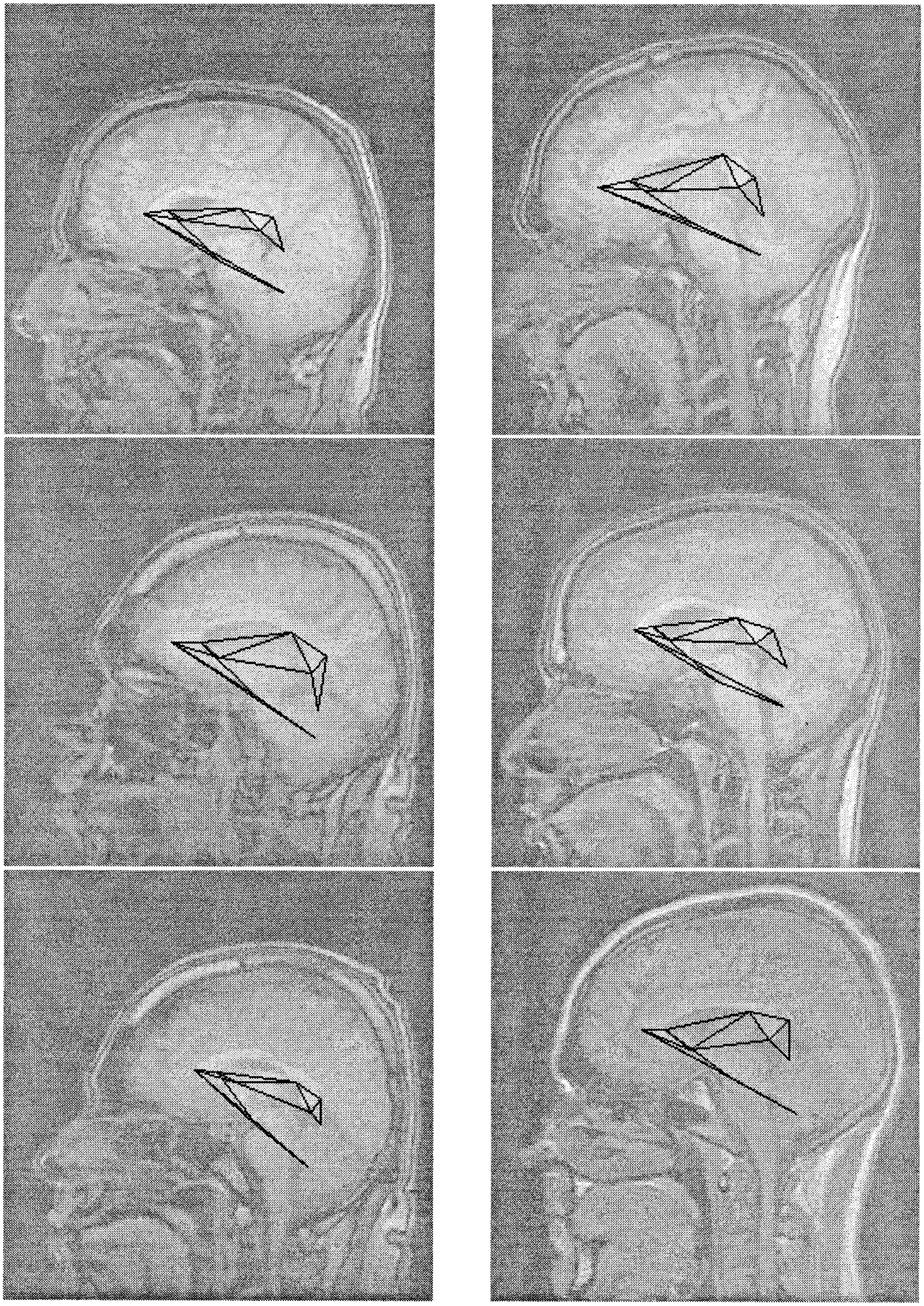

Fig. 10. Successful matches to six sagittal MRI brain images.

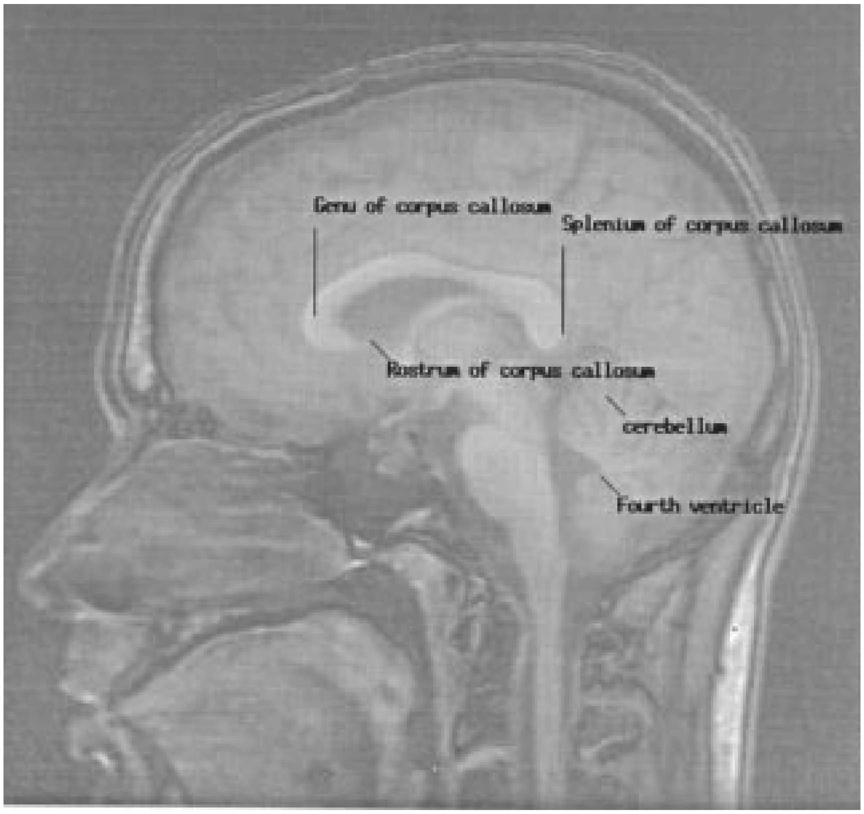

Fig. 11. Automatic identification of anatomies in a sagittal image.

TABLE I
SUMMARY OF OUTCOMES ON 204 AXIAL IMAGES

	Good images	Bad images	Total
Good match	49	0	49
Bad match	1	28	29
Rejected	7	119	126
Totals	57	147	204

of the images contained anything similar to the structures in question. This could be due to either severe abnormalities or a large variation in the angle of the MRI scans.

Good images were expected to contain the frontal and posterior horns of the ventricles, the small lobes beneath the frontal horns, and the small triangle between the posterior horns, indicating the internal cerebral vein. Among these images there was extensive variability in the shape and width of the ventricles, the angle between them, the intensity contrast between the ventricle, and the background, etc.

Using 16 axial images which were available prior to the experiment, we determined a threshold for rejection. The average global cost for these 16 images was 2.1 with lower quartile 1.25 and upper quartile 2.64. On one image the match was wrong with global cost of 4.5. The threshold of four for the global cost was, therefore, chosen. Any match below that threshold would be accepted and any match above it would be rejected.

Table I summarizes the results on the 204 images. There are three possible outcomes per good image: the global cost is below the threshold and the match is a good one (as judged by eye), the global cost is below the threshold and the match is bad, and the global cost is above the threshold and the image is rejected. For bad images only the last two outcomes are possible. It should be noted that there is a subjective aspect to judging if a match is good.

It is clear from this summary that the algorithm does rather well on good images, however it tends to accept too many bad images. The current model is therefore insufficient as a classifier between those images which do and those which do not have the relevant structure.

J. Software

Code is available at:

http://galton.uchicago.edu/~amit/Software/DYN.

It takes as input the locations of the vertices in the model, the template graph (optional) and the hard constraints on the angles (optional). If these are not provided, the default second-order linear graph is used and hard constraints of ±90° are applied. Also available is a library of Matlab functions which provide an interactive framework for choosing the landmarks in the model on a sample image, and choosing the appropriate masks. For the models presented here the typical computation time is between 10–30 s on a SUN SPARC 10, with 100 candidates on average per landmark.

III. THE BAYESIAN FORMULATION: HIDDEN MARKOV GRAPH MODELS

The graph matching algorithm can be cast in a Bayesian setting and has a very similar appearance to the well-known

Hidden Markov Models (HMM's), which are extensively used in speech recognition, and a host of other time-dependent data. In all such models an underlying Markov chain on a discrete state space generates observed data according to some stochastic mechanism. The Markov process can be considered the prior distribution on the space of state sequences, and the data generating mechanism becomes the likelihood. Dynamic programming and expectation-maximazation (EM) type algorithms are combined to estimate parameters from training data, and ultimately to find the mode of the posterior distribution. These models are tractable computationally due to the Markov nature of the prior and the fact that the data model assumes conditional independence given the state sequence.

Let $v_1, \cdots, v_n$ be the vertices of the graph, and let $\mathcal{C}$ be the collection of graph cliques, which in the specific case described above were all triples of vertices. Divide the $l \times l$ pixel grid L into disjoint $k \times k$ squares centered at the subgrid of pixels $L_s = \{x_1, \cdots, x_M\}$, where $M = l^2/k^2$. Assume that the locations of the graph match are always on the subgrid L_s. The state space at each vertex of the graph is the subgrid L_s. If $D = (D_1, \cdots, D_n)$ denotes the random n-tuple of pixel locations indexed by the graph the prior can be written as

$$P(D = d) = \frac{\exp[E(d_1, \cdots, d_n)]}{Z} = \frac{1}{Z} \exp\left[\sum_{C \in \mathcal{C}} \phi_C(d_C)\right].$$

The normalizing constant Z is given by

$$Z = \sum_{d \in S} \exp\left[\sum_{C \in \mathcal{C}} \phi_C(d_C)\right]$$

where S is the subset of all n-tuples which satisfy the hard constraints.

The data image I is defined on the lattice L. For any subset N of L, let $I_N = \{I_z : z \in N\}$ be the subimage indexed by N. The distribution of I given a certain graph configuration $d_1, \cdots, d_n$ is formulated as follows. Assume that conditioned on the graph configuration d, the data $I_{N(x_i)}, i = 1, \cdots, M$ are independent, where $N(x_i)$ is the $k \times k$ neighborhood of x_i. For any pixel $x \in L_s$ let $\mathcal{T}(I_{N(x)}) = (T_1(I_{N(x)}), \cdots, T_R(I_{N(x)})$ be a vector of local statistics depending on the image data $I_{N(x)}$ in the neighborhood $N(x)$ of x.

For each vertex in the graph write the conditional distribution of the data in the neighborhood of its location in the image as

$$\begin{aligned} &P(I_{N(x)} \mid D_j = x) \\ &\quad = P(\mathcal{T}(I_{N(x)}) \mid D_j = x) P(I_{N(x)} \mid \mathcal{T}(I_{N(x)})) \\ &\quad = F_j(\mathcal{T}(I_{N(x)})) G(I_{N(x)}) \end{aligned} \tag{2}$$

where G is a function which does not depend on j. For $x \in L_s$ the conditional distribution on the data given no vertex is located at x, namely, x is in the background, is given by

$$P(I_{N(x)} \mid x \notin \mathcal{D}) = F_0(\mathcal{T}(I_{N(x)})) G(I_{N(x)})$$

where $\mathcal{D} = \{d_1, \cdots, d_n\}$. Thus, $\mathcal{T}$ is a vector of sufficient statistics for the family of distributions $P(D_{N(x)} \mid D_j = x),\ j = 1, \cdots, n$ and $P(D_{N(x)} \mid x \notin \mathcal{D})$.

The full posterior distribution is obtained by Bayes formula as

$$\begin{aligned} P(D = d \mid I) = \frac{1}{Z} \exp&\left[\sum_{C \in \mathcal{C}} \phi_C(d_C)\right] \\ &\cdot \prod_{j=1}^{n} F_j(\mathcal{T}(I_{N(d_j)})) G(I_{N(d_j)}) \\ &\cdot \prod_{x \in L_s - \mathcal{D}} F_0(\mathcal{T}(I_{N(x)})) G(I_{N(x)}). \end{aligned} \tag{3}$$

Dividing this expression by

$$\frac{1}{Z} \prod_{x \in L_s} F_0(\mathcal{T}(I_{N(x)})) G(I_{N(x)})$$

the following objective function needs to be maximized to obtain the mode of the posterior distribution

$$\begin{aligned} J(d) &= \exp\left[\sum_{C \in \mathcal{C}} \phi_C(d_C)\right] \prod_{j=1}^{n} \frac{F_j(\mathcal{T}(I_{N(d_j)}))}{F_0(\mathcal{T}(I_{N(d_j)}))} \\ &= \exp\left[\sum_{C \in \mathcal{C}} \psi_C(d_C)\right] \end{aligned} \tag{4}$$

where

$$\begin{aligned} \psi_C(d_C) = \phi_C(d_C) + \sum_{j \in C} &\log(F_j(\mathcal{T}(I_{N(d_j)}))) \\ &- \log(F_0(\mathcal{T}(I_{N(d_j)}))). \end{aligned}$$

It is seen that the second factor in each of the conditional distributions is eliminated and hence does not need to be specified. The entire construction depends only on the vector of statistics $\mathcal{T}$. The function $\log J$ is again expressed as a sum of functions on the clique configurations and, assuming the template graph is decomposable, is amenable to optimization through dynamic programming.

In the implementation described in this paper a very crude version of this general formulation is employed. Each vertex i in the graph is assigned a statistic T_i. The probability functions F_i depend only on the component T_i (this is, of course, possible only because the state spaces are finite), moreover, these functions are constant over all configurations I_N for which $T_i(I_N) > t_i$ for some threshold t_i. The function F_0 is simply the uniform distribution over all configurations, and therefore, contributes nothing to the objective function J.

The Bayesian formulation is presented mostly for conceptual purposes. It is not very realistic to assume conditional independence of the data in the blocks given the locations of such a small number of landmarks. There are many structures in the "background" which are not explained by the model. However, this formulation actually does bring up the issue of modeling the background. Any model that is used will have to depend on the same sufficient statistic vector $\mathcal{T}$ in order to obtain the tractable form of the functional $J(d)$ in (4). The Bayesian formulation also raises the issue of the relative importance of the prior versus the likelihood term. Contrary to many other applications of Bayesian models in imaging, here, the prior is very realistic and offers a flexible tool for incorporating anatomic information.

IV. Discussion

The strength of this algorithm is its computational efficiency and the absence of any need to initialize the matching algorithm or the optimization procedure. This is one of the drawbacks of many *relaxation* techniques. If the template or model is not initially set nearby the correct location, the relaxation algorithm may end up in completely erroneous solutions. The shape variability is described in terms of a relatively small number of parameters which can either be estimated on training data or determined by users according to their prior knowledge of the subject matter.

Further research is still needed on incorporating other types of the many local operators proposed in the computer vision literature; on combining information provided by several graphs; on using these graph models to recognize the different anatomies; on merging axial and sagittal matches in order to obtain 3-D information; and, finally, on extending the entire paradigm to 3-D data.

Acknowledgment

The author would like to thank D. N. Levine of the MRI section of the University of Chicago Radiology Department, for allowing the use of data from their machines. He would also like to thank D. K.-C. Chu for all the effort he put in acquiring hundreds of images for this study.

References

[1] J. Talairach and P. Tournoux, *Co-Planar Stereotaxic Atlas of the Human Brain*. New York: Thieme Medical, 1988.

[2] Y. Amit, U. Grenander, and M. Piccioni, “Structural image restoration through deformable templates,” *J. Amer. Statistical Assoc.*, vol. 86, no. 414, pp. 376–377, 1991.

[3] Z. Jin and P. Mowforth, “A discrete approach to signal mapping,” The Turing Institute, Tech. Rep. TIRM-88036, 1988.

[4] R. Bajcsy and S. Kovacic, “Multiresolution elastic matching,” *Comput. Vision, Graphics, Image Processing*, vol. 46, pp. 1–21, 1988.

[5] M. Miller, G. Christensen, Y. Amit, and U. Grenander, “A mathematical textbook of deformable neuro-anatomies,” in *Proc. National Academy of Science*, 1993, vol. R90, pp. 11944–11948.

[6] Y. Amit, “A nonlinear variational problem for image matching,” *SIAM J. Scientific Computing*, vol. 15, no. 1, pp. 207–224, 1994.

[7] U. Grenander, Y. Chow, and D. M. Keenan, *A Pattern Theoretical Study of Biological Shape*. New York: Springer Verlag, 1991.

[8] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *Int. J. Comput. Vision*, pp. 321–331, 1987.

[9] T. F. Cootes and C. J. Taylor, “Active shape models—smart snakes,” in *Proc. BMVC*, 1992, pp. 267–275.

[10] R. R. Petrocelli, J. L. Elion, and K. M. Manbeck, “A new method for structure recognition in unsubstracted digital angiograms,” in *Proc. Computers in Cardiol.*, 1992, pp. 207–210.

[11] L. F. Bookstein, *Morphometric Tools for Landmark Data: Geometry and Biology*, Cambridge: Cambridge University Press, 1991.

[12] Y. Amit and D. Geman, “Shape quantization and recognition with randomized trees,” *Neural Computation*, to be published.

[13] Y. Amit and A. Kong, “Graphical templates for model registration,” *IEEE Pattern Anal. Machine Intell.*, vol. 18, pp. 225–236, 1996.

[14] A. Wilson, “Statistical models for shapes and deformations,” Ph.D. dissertation, Inst. Statistics and Decision Sciences, Duke Univ., Durham, NC, 1995.

[15] R. M. Haralick, and G. L. Shapiro, *Computer and Robot Vision*, vols. 1 and 2. Reading, MA: Addison Wesley, 1992.

[16] D. J. Rose, R. E. Tarjan, and G. S. Leuker, “Algorithmic aspects of vertex elimination on graphs,” *Siam J. Comput.*, pp. 266–283, 1976.

[17] C. Watson, *Basic Human Neuroanatomy, an Introductory Atlas*. Boston, MA: Little Brown, 1995.