MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. ??? July, 1995

Model Based Correspondence

Pamela Lipson and Shimon Ullman

This publication can be retrieved by anonymous ftp to publications.ai.mit.edu.

Abstract

We present a new technique to register an object model with an observed image. The technique uses the model to establish an image-to-model correspondence and, therefore, to facilitate the registration process. The model guides and constrains the matching procedure in order to reduce the inherent complexity of the registration problem and to increase the robustness and efficiency of the solution. The technique begins by roughly aligning the image and model using an overall affine transformation. It then determines correspondence estimates between a sparse set of model and image contours. The major contribution of the approach is that the technique is able to constrain the rest of the matches via global information from the model. This process can be repeated to refine the resulting correspondence. We have incorporated our technique into the linear combination object recognition scheme and have tested the entire system successfully on a variety of objects. There are three benefits to our approach. First, it is effective; experiments show that our procedure quickly converges to a solution, if one exists. Second, it is computationally simple and efficient; the use of models effectively constrains the matches and the process results in a linear solution. Finally, the procedure is robust; moderate errors in the rough alignment stage do not impair the subsequent correspondence procedure.

Copyright © Massachusetts Institute of Technology, 1995

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by National Science Foundation, contract number IRI 8900267, and in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-85-K-0124. The addresses of the authors are : Pamela Lipson, NE43-739, MIT AI Laboratory, 545 Technology Square, Cambridge, MA 02139, USA. Email: lipson@ai.mit.edu. Shimon Ullman, Dept. of Applied Math and Computer Science, Weizmann Institute of Science, Rehovot, 76100, Israel. Email: shimon@wisdom.weizmann.ac.il

1 Introduction

The problem we examine here is the registration of an three-dimensional object model with a two-dimensional image. This process is useful for model based object recognition as well as a variety of other visual applications including alignment of a model to data from different types of sensors.

There are two general approaches for computing model to image registration. The first searches the space of possible transformations that will align the model with an observed image. Since the space of possible transformations is large, work in this area has focused on efficient search techniques [4, 27]. A second approach tries to determine corresponding pairs of image and model features that may be used to recover a transformation which maps the model onto the image. Some examples are [8, 15, 17, 19]. Feature correspondences are computed in a "bottom up" manner, where image features, mainly pointwise, are matched with model features. By "bottom up" pairing of features we mean that potential matching features in the image are selected independently of the stored object model. Transformations based on these matches are computed and the result is evaluated. To reduce complexity of the possible feature correspondences, "bottom up" techniques focus on developing heuristic rules to eliminate incorrect pairings and therefore the number of transformations computed. In this paper we describe a method for registration that uses the model or "top down" information to determine the feature correspondences.

There are two disadvantages to "bottom up" approaches. The first has to do with the ambiguity of the matches. Given a feature in the image, such as a corner or inflection point, there are often multiple potential matching features in the model (see figure 1). The second problem is that most previous methods use point like features that are well localized in the image plane, but in many images robust features of this type are rare.

In the model-based method proposed in this paper these two problems are significantly reduced. The method combines the use of pointwise and extended contours in the matching process and it removes the matching ambiguity by using information concerning the possible transformations of the model.

The remainder of this paper presents the model-guided approach to registration within the framework of a model-based recognition system. The particular system implemented uses view-based models, contours as features, and a linear combination recognition engine. Section 2 discusses the motivation behind the choice of pictorial models and contour features, and argues that the model-based approach to correspondence is especially effective in these domains. Section 3 presents an overview of the model-guided contour matching algorithm. Section 4 describes the specific implementation. Section 5 shows examples of the approach tested on a range of synthetic and natural models and images within a recognition application.

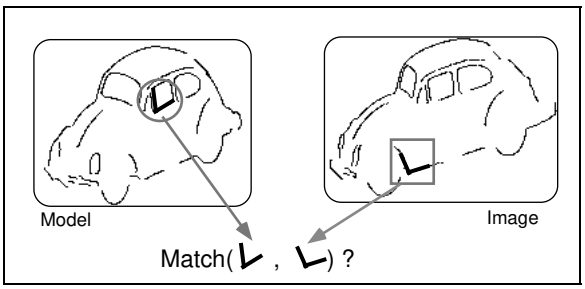

Figure 1: *One of the disadvantages of using only "bottom up" information in the matching process. Two pictures of a VW are shown. One represents the model and the other the image. The goal is to find a correspondence between image and model features. One model corner feature and one image corner feature are highlighted. Looking at these two features in isolation, or in a "bottom up" manner, it is difficult to decide whether these are a matching pair.*

2 Model and Image Representations

A technique that compares an incoming image with a stored model requires decisions regarding the representation of the image and the object. Common approaches use grey level primitives, edge representation, surfaces, and volumetric primitives.

The implementation of the model-guided technique represents objects by a small number of corresponding two-dimensional views (see figure 2). This modeling strategy is known as the pictorial approach. The benefits of incorporating pictorial models into a registration scheme are: 1) the models are simple to generate and store and 2) that they are in the same format as the images - thus, facilitating the model-image match process. In addition, recent psychophysical evidence suggests that the human visual system recognizes images by matching them with previously stored two-dimensional views (without the use of explicit three-dimensional representations) [3, 5, 6].

Our technique represents each view as a set of contours. The representation can also be augmented with information about special feature points such as corners or inflection points. The matching is performed between contours derived from the image and contours in the stored object model.

Contours satisfy many of the requirements for good image and model feature representations (see figure 3). A benefits of using contours is that they can be viewed as local features and also seen as a global arrangement. They also allow for some invariance to lighting and changes in viewpoint. Our definition of "contours" parallels the intuition of their being collections of connected points. Given this definition, contours are easy to generate, via a rudimentary edge-detection processing step. (No further abstraction beyond edge-detection is required.) Additionally, they are simple to store as image data.

Despite these advantages, contours have been largely neglected as a possible feature representations. Unlike pointwise features, the matching of extended contours is

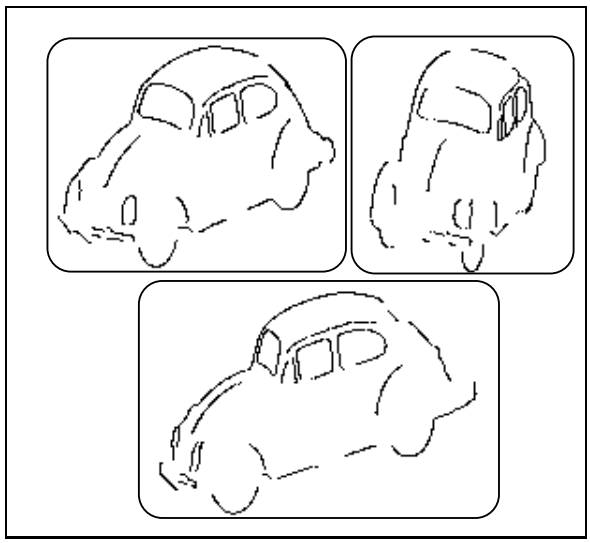

Figure 2: *Example of a pictorial model of a VW.*

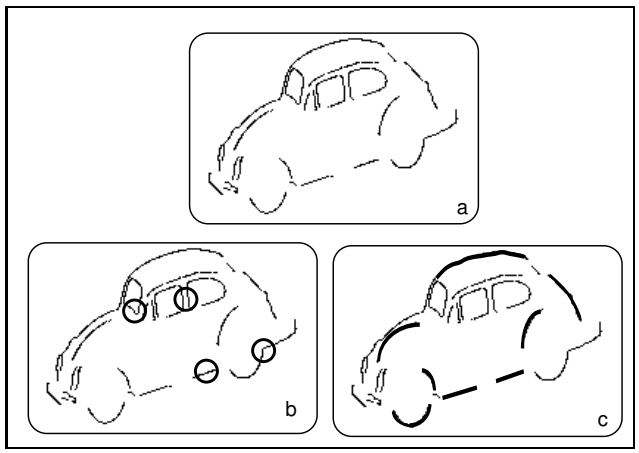

Figure 3: *Representations for Features. (a) A real image. (b) Most methods to determine correspondence attempt to extract explicit features from images like this one and then match them to their equivalents in models. Some examples of local explicit features are corners, vertices (where several edges meet), blobs, cusps, and inflection points. Object parts and centers of salient features typify more global explicit features [25]. Some of these explicit features are highlighted here (c) Looking again at the explicit features in part (b), it is obvious that a lot of information in the image is lost in the translation process from raw image data to explicit features. In particular, the curved segments, which constitute a large majority of images, are ignored. The curves, however, can provide a wealth of data to the correspondence process. A few contours that could be used as features are highlighted.*

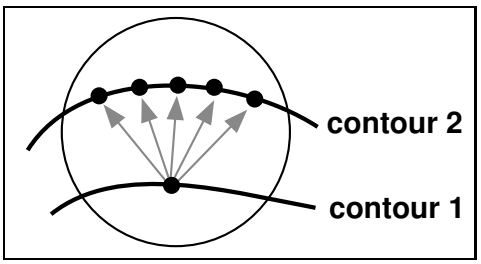

Figure 4: *The aperture problem - Looking through a local aperture a point on contour 1 may match with any of an infinite number of points on contour 2. Only the perpendicular component of the match can be recovered [18]*

still ambiguous as to the exact matches between individual points on the two contours. This problem is similar to the "aperture problem" in motion correspondence (see figure 4).

In motion computation, the ambiguity problem is usually approached by using general constraints such as smoothness or assumptions of local affine motion fields [1, 12]. Such approaches try to find the point-to-point match that maximizes these constraints. These techniques can be thought of as a "bottom up" processes, because they do not use information associated with specific models.

The use of models in the matching of image and model contours provides valuable information to reduce the ambiguity of the problem. The model provides information regarding the arrangement or structure of the contours. Anchor points in both the model and the image can bring the two into close registration. Most importantly, the model provides the information necessary to compute an analog of the "normal component" of motion *and* provides constraints to determine a fine correspondence between contours.

3 Overall Structure of the Algorithm

In our work we attempt to demonstrate that the model ("top down data"), in conjunction with "bottom up" data, can be used effectively to solve the correspondence problem. Traditional "bottom up approaches" may have to match all image features to all possible model features before the search for a consistent pairing is terminated. A more efficient algorithm for computing correspondence is to first to roughly distort the image to match the model to constrain the possible search space, then to get a rough match of a minimal set of features in a "bottom up" fashion, and finally, in the main part of the algorithm, to use the model to constrain and refine the rest of the matches. The reduction in complexity of the problem through such an approach is illustrated in figure 5.

Our proposed algorithm for computing correspondence between a model and an image is summarized pictorially in figure 6. The inputs to the system are a set of two-dimensional model pictures and an image. We use

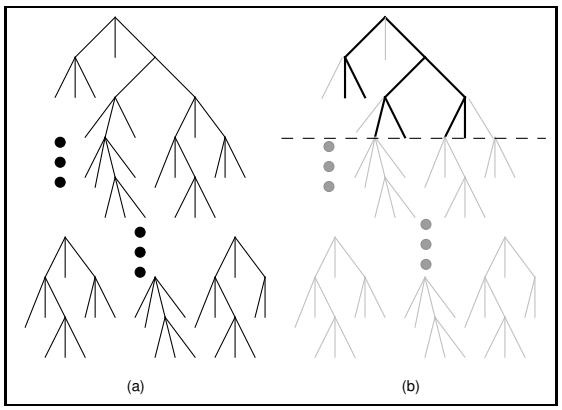

Figure 5: *The advantages of using the model to guide the correspondence process over a purely "bottom-up" approach. (a) The "bottom" up approach may have to evaluate all the possible pairings of image and model features. The graphic depicts a search tree where every model feature is compared to every image feature. The leaves of the tree represent the set of all the possible image-model feature pairings (b) Our technique uses the model to constrain the complexity in two ways. Looking at the tree above the horizontal dotted line, we reduce the search for a few initial correspondences by roughly distorting the model to match the image. The rest of the search tree below the dotted line is eliminated as we employ "top down" information to constrain and refine the rest of the feature correspondences without any additional search.*

contours as both image and model features. The correspondence algorithm proceeds in four distinct steps:

1. *Rough alignment.* We roughly align one of the model pictures to the image, bringing the model and image features into close spatial registration. The decision of which model picture to pick can either be arbitrary or be guided by some simple criteria such as similarity of dominant orientations between the model picture and the image. The result is a new image which contains a superimposition of the transformed model and the image. Experimentally, we have found that a few simple techniques can be used effectively to achieve this coarse registration.

2. *The initial matching of image and model contours.* We first compute a small number of "bottom-up" correspondences between contour points from the model picture and the image. However, because of the aperture problem, described in the previous section, we cannot specify an exact match between contour points. Therefore, a range of possible matches is given to the next step for refinement.

3. *Model-guided matching.* This step is the main contribution of the algorithm. The model-guided matching or fine alignment stage relies heavily on information from the model to produce exact contour matches and to achieve a full point-to-point correspondence. We incorporate some ideas from recovery of optical flow to implement this procedure. The basic premise is to match image and model curves using local tangential constraint

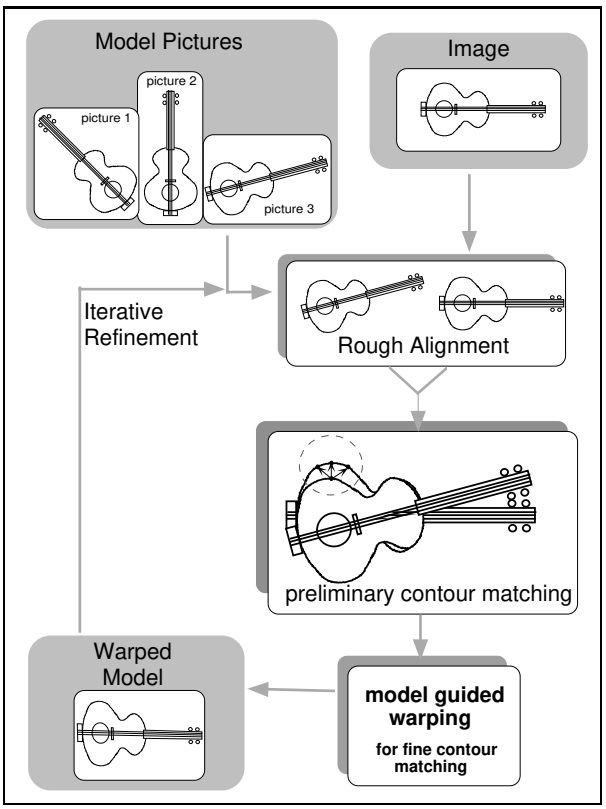

Figure 6: *Summary of the proposed correspondence process. The process begins by roughly aligning the image with a model picture. We then postulate rough matches between a few model and image contour points in a "bottom up" fashion. These rough matches are sent to the next stage where they are refined using constraints from the model. Using these matches, we are able compute a transformation that maps the model features onto the image features. The transformed or warped model is compared to the image. We can refine the transformation by feeding the warped model back into the process as one of the model pictures.*

lines and global model information. In this step, the contours are matched simultaneously to achieve a stable, globally consistent solution.

4. *Verification and iterative refinement.* The resulting correspondence is used to determine a transformation that takes the model and brings it into alignment with the image. A verification stage determines the validity of the transformation. It is possible to further refine the transformation by repeating the whole process again using the warped model and the image as inputs.

In section 4, we describe the details of our correspondence algorithm.

3.1 Assumptions

In our implementation, we make several assumptions about the contents of the image and the model used in the recognition process. First, we assume that segmentation has already been performed, meaning that there is only one object in the image, and that the image was generated under orthographic projection. Also, our algorithm utilizes two-dimensional edge data. Therefore, both the image and model pictures are assumed to contain only contours. Furthermore, to ensure having a reasonably detailed model of an object, pictures which comprise the model are assumed to be taken in intervals of 30 degrees or less. We also assume that we know the correspondence between points in different model images. This is not an unreasonable assumption given the fact that we construct the models off-line before the start of the recognition process.

3.2 The Linear Combination Approach to Recognition

We have chosen to test our model-driven approach to correspondence by embedding it in the linear combination of views framework for object recognition [26].

Ullman and Basri showed that one can model all the possible views of an object by a linear combination of a small number of fixed two-dimensional views, $\mathbf{m_1}, \mathbf{m_2}, ...\mathbf{m_n}$, of the same object. Given an observed image, $\hat{\mathbf{p}}$, the goal of the recognition procedure is to find coefficients, α_i, for each of the pictures in the model, such that the result of the coefficients applied to the model pictures approximates the image (see equation 1). If such coefficients exist, the observed image is recognized as an instance of the model.

The coefficients for the linear combination encode the three-dimensional rotation, translation, and scaling transformations that map a model onto a viewed object. The process of recovering these transformations, mapping the model into image space, and comparing the result with the image is known as alignment.

$$\hat{\mathbf{p}} = \alpha_1 \mathbf{m_1} + \alpha_2 \mathbf{m_2} + \cdots + \alpha_n \mathbf{m_n} \qquad (1)$$

The key problem here is to find a correspondence between the model and image features. If a pointwise correspondence is known, it is relatively simple to determine the alignment coefficients by solving an overdetermined set of linear equations. Ullman and Basri showed that within this framework it is possible to use only a small number of corresponding points to align a model of pictures with an observed image [26].

4 Model-Guided Correspondence – Description of the Algorithm

We now describe our proposal for solving the correspondence problem. As mentioned in section 3 it has four parts: rough alignment, initial matching of image and model contours, model-guided matching, and verification and iterative refinement. The third step describes the novel use of the model in the registration process and is the focus of the algorithm.

4.1 Rough Alignment

The correspondence process begins by partially compensating for gross changes in the viewing geometry between an image and a model. In the ideal case, the rough alignment should compensate for differences in in-plane translations, in-plane rotations, in-depth displacements, small in-depth rotations, and small non-rigid transformations [25]. However, a compensatory step which does not bring the image and model into accurate alignment, but only reduces their overall difference, can be very useful. We used a combination of simple existing techniques to achieve this latter goal. Some of these methods are detailed in [14, 23].

The rough alignment stage is composed of four steps:

1. *Compensating for translations in xy plane.* The matching of a single identifiable feature in the image and the model can be used to compensate for translation. One feature which can be used effectively for this purpose is the center of mass (assuming no background clutter or occlusion) We used first moments to calculate the centers of mass of the image and model pictures.

2. *Compensating for rotations in xy plane.* Alignment of the dominant orientations of the figures in the image and model pictures can compensate for differences in in-plane rotations. Computing the axis of least inertia of the points in a picture will approximate the dominant orientation of the figure. However, the axis of least inertia does not specify a unique direction for the dominant orientation. Some global clues to eliminate the two-way ambiguity are symmetry and distribution of mass. More local components such as tangential points, blobs and local texture can be integrated into the process.

3. *Scaling in the x and y directions.* Scaling in the x and y directions provides partial compensation for small rotations in depth. A full compensation is achieved in the later stage of model-guided matching.

4. *Overall scaling.* Overall scaling of the image or model can approximate translations in depth. We can measure the overall difference in size between the image and model figures. (Some measures of difference include comparing the dimension of greater magnitude between the image figure and model figure or comparing the average x and y dimensions of the boxes bounding the image figure and model figure.) The image figure can be scaled uniformly in both dimensions to compensate for this difference.

After these transformations, the image and the model

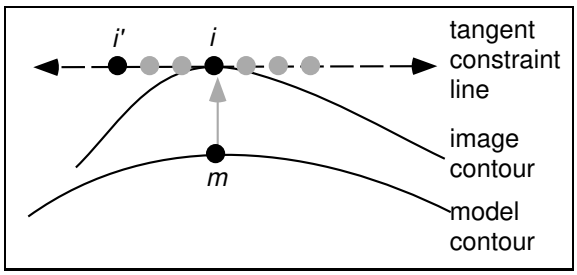

Figure 7: *Contours are represented by local tangent lines. The match to a model contour point, m, is either the radially closest image point, i, or a point, i', along the contour-tangent at i.*

should be in good but not perfect alignment. The subsequent correspondence stages produce a full compensation.

4.2 Initial Matching of Image and Model Contours

The goal of this stage is to estimate correspondences between a small number of model and image contours. Let $\mathbf{m}$ be the model picture that, based on a least sum of squares error analysis, most resembles the observed image, $\hat{\mathbf{p}}$. Assume $\mathbf{m}$ and $\hat{\mathbf{p}}$ have been roughly aligned and that we have a new "work" image containing a superimposition of the two. This stage approaches the matching of contours in a "bottom up" fashion. The output is a set of possible matching points between a small number of image and model contours.

The process of matching contours begins with choosing points which lie on contours in $\mathbf{m}$. A sparse number of points are chosen by a sampling of all the model points. Based on proximity criterion we match each chosen chosen model contour point m with its corresponding point on an image contour. For each of these points m, using a radial search, we produce a candidate i, where i is the closest image point to the model point m. Given that we have already roughly aligned the image and model, the true corresponding point to m is expected to be either i or a point in the neighborhood of i along the image contour containing i.

We approximate this constraint by requiring the match of point m to lie on the tangent to the curve at i (see figure 7). This constraint has been used in other contexts such as the recovery optical flow [11]. The accuracy of this constraint depends on the curvature at i and the distance of the correct matching point from i. As we shall see in the examples below, these deviations tend to average out, and accurate matches are obtained even when contours with significant curvature are used.

It is the goal of the subsequent stage to condense the guess from all the points on the tangent to a image contour tangent point i' that is closest to the true corresponding image point. (Note that although the final decision may not be a point lying on the image contour, the procedure should choose the tangent point which is closest to the real corresponding match.)

Let v_x and v_y be respectively the x and y components of the tangent vector to the local image curve at i. Let t be the distance between i and i' along the tangent vector. The tangential assumption states that

$$(i'_x, i'_y) - (i_x, i_y) = t(v_x, v_y) \tag{2}$$

t can be removed from the equation 2. The result is the equality

$$v_y i'_x - v_x i'_y = v_y i_x - v_x i_y \tag{3}$$

The only unknowns here are (i'_x, i'_y). It is the goal of the next stage to resolve these unknowns using the local tangent constraints and the global model constraints.

4.3 Model-Guided Matching

The model-guided matching stage constitutes the main part of the registration algorithm. It uses model information to resolve the contour matches in order to produce a detailed pointwise correspondence. This stage simultaneously picks a unique correspondence for each chosen model point via model based constraints. The final solution is the one that best satisfies these constraints (see figure 8).

According to the linear combination approach, a new image of an object can be represented as a linear combination of known (model) pictures of the same object (see equation 1). Equivalently, the coordinates of a point on that image can be calculated by taking a linear combination of the coordinates of its corresponding model points. Thus, given an image point i', if $m_1, m_2, m_3, ..., m_n$ are corresponding points in model pictures 1 through n respectively, then the following equations are valid.

$$i'_x = \alpha_{1x} m_{1x} + \alpha_{2x} m_{2x} + \cdots + \alpha_{nx} m_{nx} \tag{4}$$

$$i'_y = \alpha_{1y} m_{1y} + \alpha_{2y} m_{2y} + \cdots + \alpha_{ny} m_{ny} \tag{5}$$

This model based set of constraints is used to recover the coordinates for i'.

In the last section, i' was constrained to lie along a tangent line to the image contour at point i. The right hand sides of equations 4 and 5 can be substituted for (i'_x, i'_y) in equation 3 to obtain equation 6.

$$\begin{aligned} v_y i_x - v_x i_y = \\ v_y(\alpha_{1x} m_{1x} + \alpha_{2x} m_{2x} + \cdots + \alpha_{nx} m_{nx}) - \\ v_x(\alpha_{1y} m_{1y} + \alpha_{2y} m_{2y} + \cdots + \alpha_{ny} m_{ny}) \end{aligned} \tag{6}$$

Now the only unknowns are the linear combination coefficients $(\alpha_{1x}, \alpha_{2x}, ..., \alpha_{nx})$ and $(\alpha_{1y}, \alpha_{2y}, ..., \alpha_{ny})$. The correspondence problem, thus, is reduced to solving for these coefficients such that they satisfy equation 6. Once the coefficients are recovered, equations 4 and 5 can be used to determine the coordinates of i', the final goal.

The model-guided technique, which determines $(\alpha_{1x}, \alpha_{2x}, ..., \alpha_{nx})$ and $(\alpha_{1y}, \alpha_{2y}, ..., \alpha_{ny})$, generates more than $2n$ equations of the form of equation 6. There is one equation for each of the chosen model contour points in $\mathbf{m}$. The result is an overdetermined set of linear equations.

The following is equation 6 in matrix form.

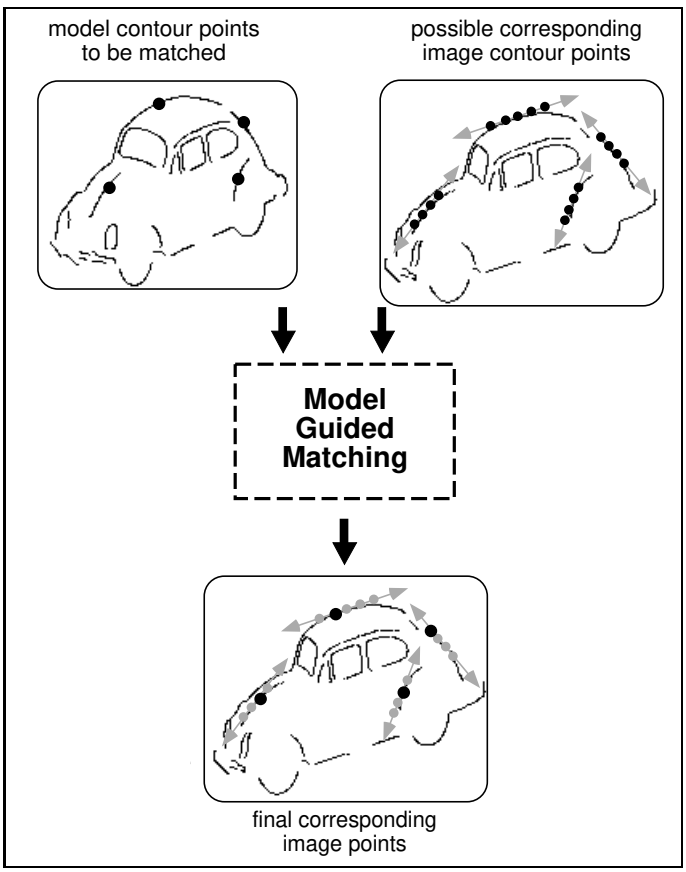

Figure 8: *The fine alignment stage takes as input the set of possible image contour points that match the selected model contour points. It then uses model guided constraints to pick a unique correspondence for each model point.*

$$\begin{pmatrix} m_{1x}v_y \ldots m_{nx}v_y & -m_{1y}v_x \ldots - m_{ny}v_x \end{pmatrix} \begin{pmatrix} \alpha_{1x} \\ \vdots \\ \alpha_{nx} \\ \alpha_{1y} \\ \vdots \\ \alpha_{ny} \end{pmatrix} = v_y i_x - v_x i_y \quad (7)$$

This set of linear equations can be solved in the least square sense using the pseudo-inverse, which calculates the best average solution for the unknowns [20, 24]. An alternative solution is to use a voting technique, such as the hough transform, which would provide the most consistent coefficients for the majority of the equations [2, 7]. (Note that the general registration technique is flexible such that the matching of pointwise features can be combined with the matching of extended contours.)

Once the linear combination coefficients that transform the model onto the observed image are recovered, we can determine a detailed pointwise model to image correspondence.

4.4 Verification and Iterative Refinement

The hypothesis, described by the coefficients recovered in the model-guided matching stage, is verified by computing the linear combination of the model pictures and then comparing the result with the observed image. An estimate of tolerance for error in the mapping, allows the procedure to determine if solution is correct.

One of the benefits of the algorithm is that the reconstructed picture, produced by the linear combination of model pictures, in addition to the image can be fed back into the process and new coefficients determined. The reconstructed picture takes the place of **m**, the initial chosen model picture . The reconstructed picture can be compared to the image using the same techniques described in sections 4.1, 4.2, and 4.3. Empirical tests have shown that if the image is truly an instance of the model, repeated refinements of the coefficients converge quickly to an improved solution.

4.5 Sources of Error

An analysis of the correspondence technique shows two potential causes of error during the "bottom up" matching of image and model features. Errors usually come at places where the model and image curves violate the local spatial proximity assumption and also at contour points of high curvature.

In the first case, it is possible for a large displacement between model and image contours to remain past the rough alignment stage. Figure 9a depicts the scenario where contour points from the model are incorrectly matched to contour points in an image because of an uncorrected large displacement. Empirically, we found that the effect of outliers often disappears in subsequent iterations of the registration process (see figure 10). Through the iterations the local spatial proximity assumption becomes valid for an increasing number of image and model contours. To make the process more robust and to decrease the number of iterations, it is possible to incorporate more information into the contour correspondence process. One possibility is to look for the sign of the contrast along the contour or to use information about whether contours are labeled as interior or boundary curves as matching criteria. In addition, outliers of this type can be eliminated by using a hough transform technique in solving equation 7 [8, 22].

Figure 9b depicts the problem arising from high curvature contours. Here the model contour point m is being compared to a high curvature image contour. Although the corresponding image point should be at the image contour peak, the local tangent line at the closest image point does not come close to this true corresponding point. One way to avoid this error is to ignore the high curvature contour areas. Because the system is already highly overdetermined, it is possible to ignore the problematic contours without affecting the performance of the correspondence technique.

5 Results

Results of the correspondence computation can be evaluated in terms of accuracy and speed. The method was

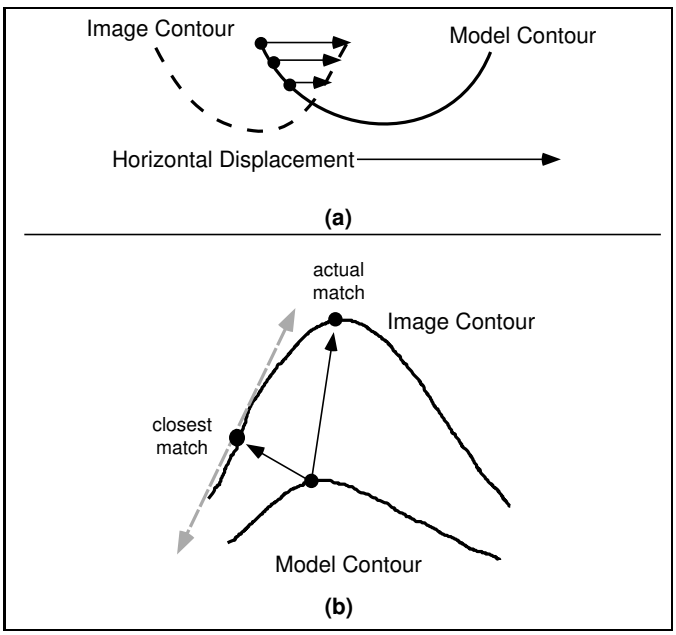

Figure 9: *Two potential sources of error in the "bottom up" feature matching stage: (a) errors due to large displacements between the image and model contours (b) and errors due to high curvature contours.*

implemented in C on a Sun Sparc IPC workstation. The procedure was tested on two-dimensional polygons, two-dimensional closed curves, synthetic three dimensional polyhedra, and real imagery. All images were binary contour images. The models were in the same format. The models consisted of two pictures for the synthetic two-dimensional objects, three pictures for the synthetic three-dimensional objects, three pictures for natural objects rotated around one axis, and five pictures for natural objects depicted in different three-dimensional rotations. For each of these models one more picture was included to compensate for translational transformations (see [26] for an explanation of the number of images that makeup a model).

The model pictures each contained between 1000-2000 contour points. Typically, the number of points used in the initial matching phase (see section 4.2) ranged from 50 points for the some synthetic objects to 100 points for the real imagery.

We used the pseudo-inverse technique to compute the linear combination coefficients. A quantitative measure of the goodness of the computed results is provided by the mean squared error between the input image and the transformed model.

In each of the following examples, we computed correspondences between a model and a novel image. We tested the procedure on novel images generated by modeled objects as well as images that came from different objects. Each of the example figures show the set of model pictures, the novel image, the novel image superimposed on one of the model pictures before the matching process, the warped model superimposed on the input image after one iteration, and, after a number of iterations, the final warped model picture superimposed on the novel image. If the novel image is an instance of the model, the final warped model picture and the novel image should be very similar.

5.1 Results with Synthetic and Real Imagery

5.1.1 Synthetic Polyhedra

Figure 11 shows the results of our correspondence process on a synthetic three-dimensional hand. The model contains three pictures of the hand in different poses as a result of rotation around one axis. The pose of the novel image is half way between the pose of the second model picture m_2 and third model picture m_3. To fully test the system, we used the most dissimilar model picture to the image, model picture m_1, to compute the contour matches. 68 model points were used in the matching process. Figure 11a shows the novel image. Figure 11b shows the image and model picture m_1 roughly aligned and superimposed. During the initial matching process, some model contours were matched with incorrect image counterparts. With subsequent iterations, the model was brought into closer registration with the image and the number of incorrect contour matches was reduced and finally eliminated. Figure 11c contains the result after the first contour matching iteration. Here the transformed model is superimposed on the image. Finally, Figure 11d shows the transformed model superimposed on the image after 3 iterations of the correspondence process. The transformed model and image are visually indistinguishable. The average squared error was 0.25 pixels/ model contour point.

5.1.2 Real Imagery

We used real pictures of a miniature Volkswagen, a miniature Saab, and a human face to further test our correspondence technique. Figures 12 and 13 are examples of the ability of our system to compute correspondences with real imagery. In both examples, initially some model and image contours were incorrectly matched. Iterations of the correspondence process, driven by "correct" contour matches, were able to bring the transformed model and image into close registration. With the final computed correspondences, our system was able to recognize new images of the cars and faces using only a few prestored pictures.

In figure 12, a toy VW was rotated around the y axis to create three model pictures. The VW model was matched with a real novel image of the car. The novel view is shown figure 12a. Note that the novel view contains many more features than the model images. Correspondence was computed between model picture m_2 and the novel image. Figure 12b shows the image roughly aligned and superimposed on model image m_2. Figure 12c shows the transformed model superimposed on the novel image after 1 iteration. Figure 12d shows the final registration after 13 iterations. The model-guided matching technique was able to register the model to the novel image with a average squared error of 1.8 pixels/model contour point.

The five model pictures in figure 13 show a human

face (KKS) under various three-dimensional rotations. The novel image shown in 13a is a new image of the face whose pose lies between the pose of the face in model pictures m_2 and m_3. For the first iteration of the correspondence process, we compare the image to model picture m_4. We specifically chose this model picture for the matching because it was quite dissimilar to the image. Figure 13b shows the comparison of the image to model picture 4. Figure 13c shows the result of the first iteration of correspondence process. Figure 13d shows the final result; the transformed model after 10 iterations superimposed on the novel image. Even though the new image is not precisely a linear combination of the model pictures, we were able to compute a reasonable model to image transformation. Our procedure recovered coefficients that correctly expressed that the novel image ($\hat{\mathbf{p}}$) was a combination of model pictures 2 and 3 (see equation 8 for the exact coefficients). Using a sum of squares error analysis, we found that the average displacement per pixel between the novel image and the transformed model was small; 2.2 pixels/model contour point.

$$\begin{aligned} \hat{\mathbf{p}}_\mathbf{x} &= 0.91\mathbf{m}_{\mathbf{2_x}} + 0.14\mathbf{m}_{\mathbf{3_x}} \\ \hat{\mathbf{p}}_\mathbf{y} &= 0.63\mathbf{m}_{\mathbf{2_y}} + 0.34\mathbf{m}_{\mathbf{3_y}} \end{aligned} \tag{8}$$

We also compared a novel image of a different face (DB) to our original face model (KKS). Figure 14 shows the results of this experiment. As expected, our correspondence technique was unable obtain a good transformation of the face model of KKS to the novel image of DB. Figure 14e shows the transformed model after 10 iterations of the matching process. The average squared error per pixel between corresponding image and transformed model points was 4.7 pixels/model contour point; more than twice the error than in the previous experiment.

We also compared images of our toy cars to the face model and images of faces to the car models. In these cases, the images had few features in common with the models. Although, the model-guided registration process tried determine a correspondence between the images and the models, usually concentrating on registering the boundary curves, the models were grossly transformed and the resulting fit was extremely poor.

5.2 Results with Inaccurate Rough Alignment

Rough alignment is composed of compensations for translation, scale, and orientation. Mistakes in the rough alignment can occur, for instance, if the image or model pictures have no discernible dominant orientations or if we cannot resolve the two-way orientation ambiguity problem. Experimentally, we found that our method can compensate quite accurately for incorrect compensations in translation and scale. However, grossly inaccurate registration of orientation is more difficult to compensate for. Psychophysical studies have suggested that large changes in orientation also hamper human recognition [21].

The subsequent example uses an image and a model picture that differ by a 18 rotation in the $x-y$ plane. In this example, initially 29 percent of the model and image contour points were incorrectly matched. The system required 13 refinements before converging to a final solution. On average, the limit in our examples was 24 degrees of in-plane rotation.

Figure 15 shows a Saab model created from various three-dimensional rotations of the car. In this case, the image is an in-plane rotation of model picture m_1 by 18 degrees. The inputs to the correspondence process are model picture m_1 and the image without any in-plane rotation compensation. Figures 15c and 15d show a comparison of the transformed model and the image after first and 13th iterations respectively. By the thirteenth iteration, the process has converged to a reasonably good solution.

5.3 Model guided matching using pointwise matches

6 Conclusion

Our goal was to create an robust and efficient technique which could compute a model to image registration. We have proposed a scheme to achieve this by using model guided information as an essential part of the matching procedure. We demonstrated our approach within a model-guided recognition application and showed the results of the process on a variety of synthetic and real objects. There are three benefits to our approach. First, it is effective; experimentally we have found that in most cases iterative refinement of the process quickly converges to a good solution, if one exists. Second, it is computationally simple and efficient; the use of models constrains the matches and allows for a linear solution. Finally, the procedure is robust; even significant errors in the rough alignment stage do not impair the subsequent correspondence procedure.

Acknowledgments

The authors would like to thank Ronen Basri, David Beymer, Aparna Lakshmi Ratan, and Greg Klanderman for their help in generating the object models and novel images. We would also like to acknowledge Ivan Bachelder for his help with the graphical user interface. Finally, we would like to thank Pawan Sinha for his helpful comments regarding this work.

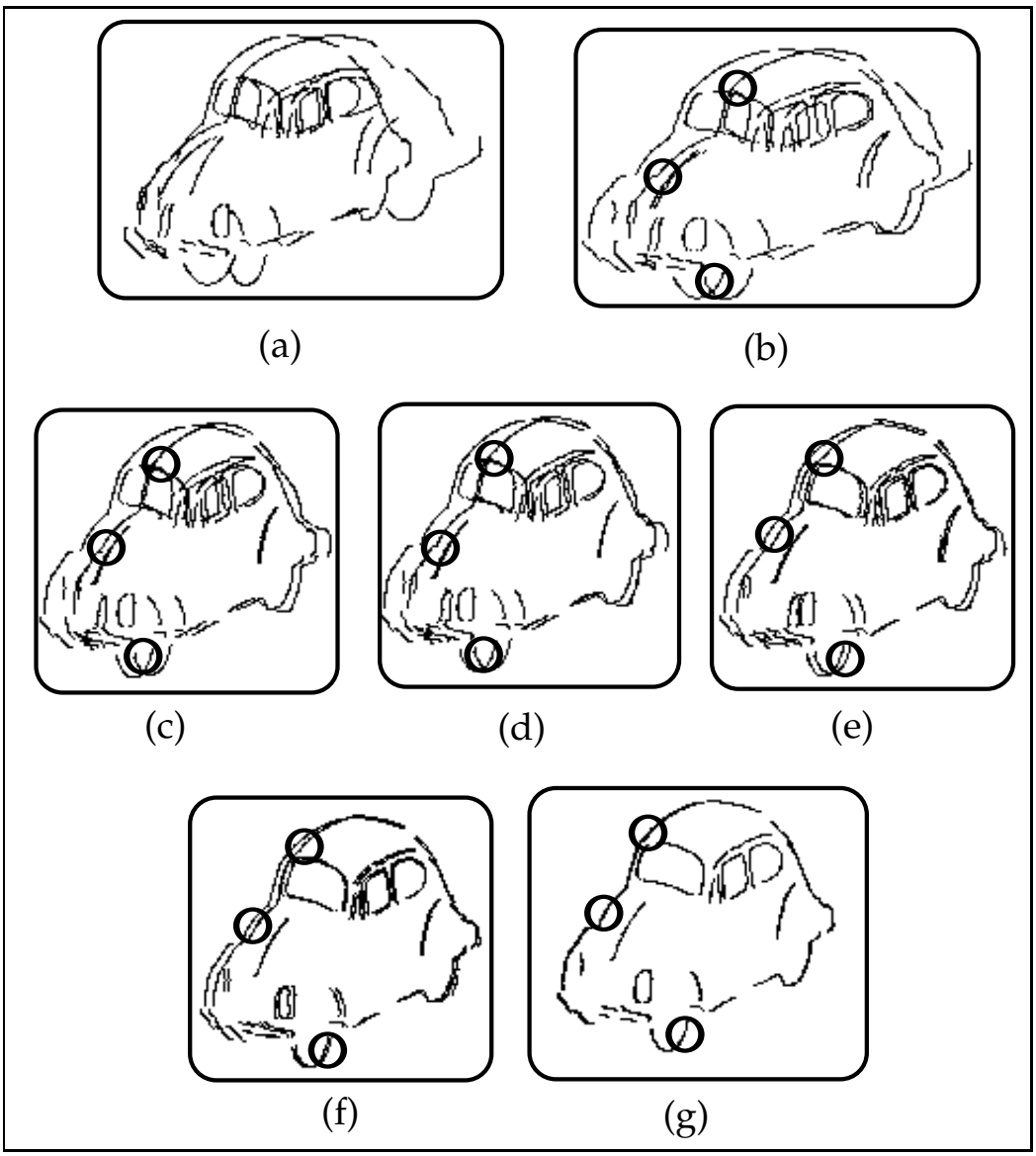

Figure 10: *Demonstration of compensation for outliers (a) Superimposition of a model and image of a VW. (b) The model and image roughly aligned and superimposed. This is the initial input to the system. The goal is to register the model with the image. The circles highlight three model and image contours which, based on the proximity assumption, are initially incorrectly matched. (b)-(g) shows actual results of the correspondence process through 5 iterations. The transformed model is always superimposed on the image. In (g) the transformed model and image are brought into correct registration, illustrating that the system can tolerate some initial incorrect contour matches*

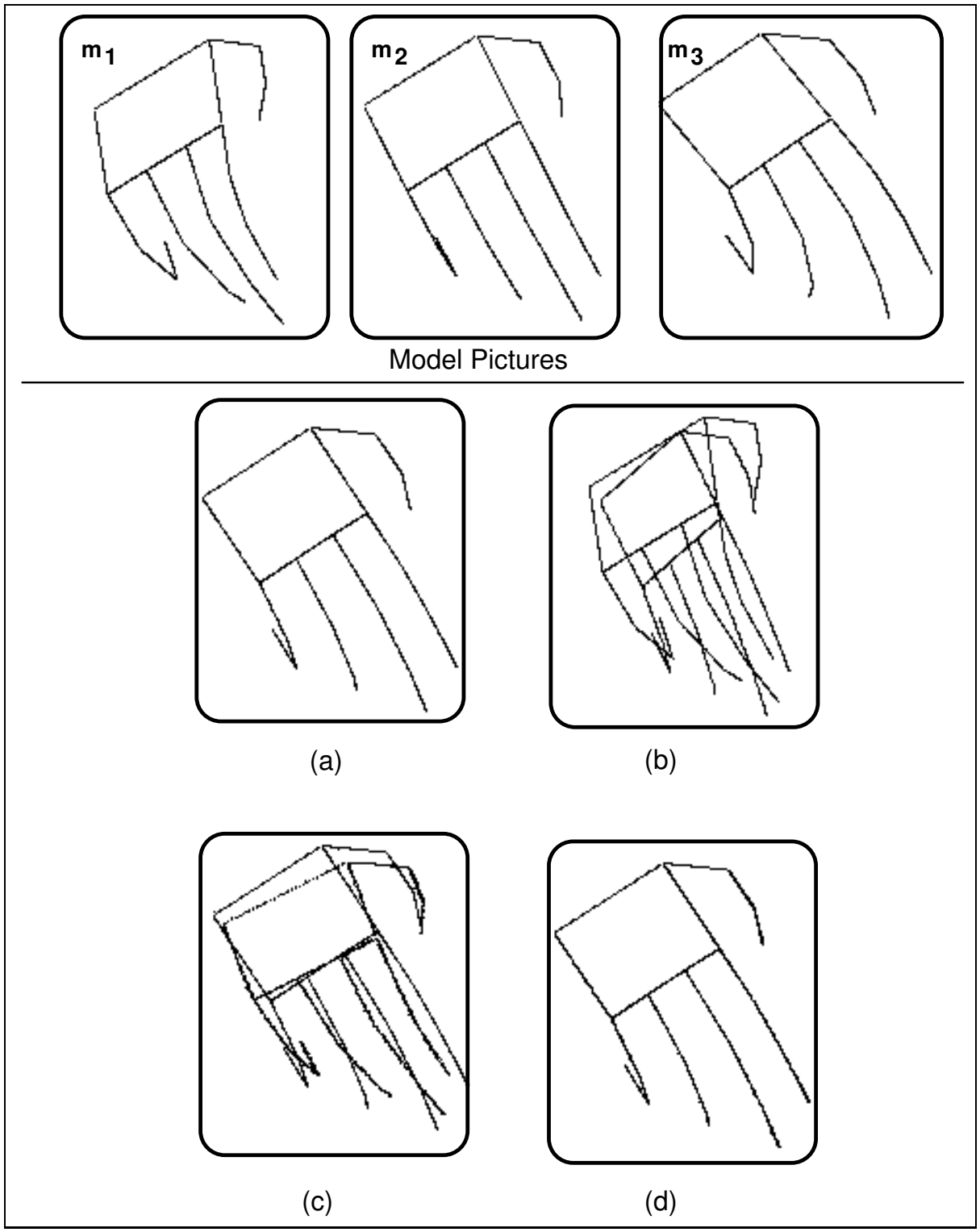

Figure 11: *Results of the correspondence technique on a three-dimensional synthetic hand. The top three pictures comprise the model. (a) The novel image (*$0.5\mathbf{m_2}+0.5\mathbf{m_3}$*). (b) The system matched the novel image to model picture* m_1*. The two are shown here roughly aligned and superimposed (c) The result of the matching process after 1 iteration; the transformed model is superimposed on the original image. (d) The final result of the matching process after 3 iterations; the transformed model is superimposed on the original image. The two are visually indistinguishable.*

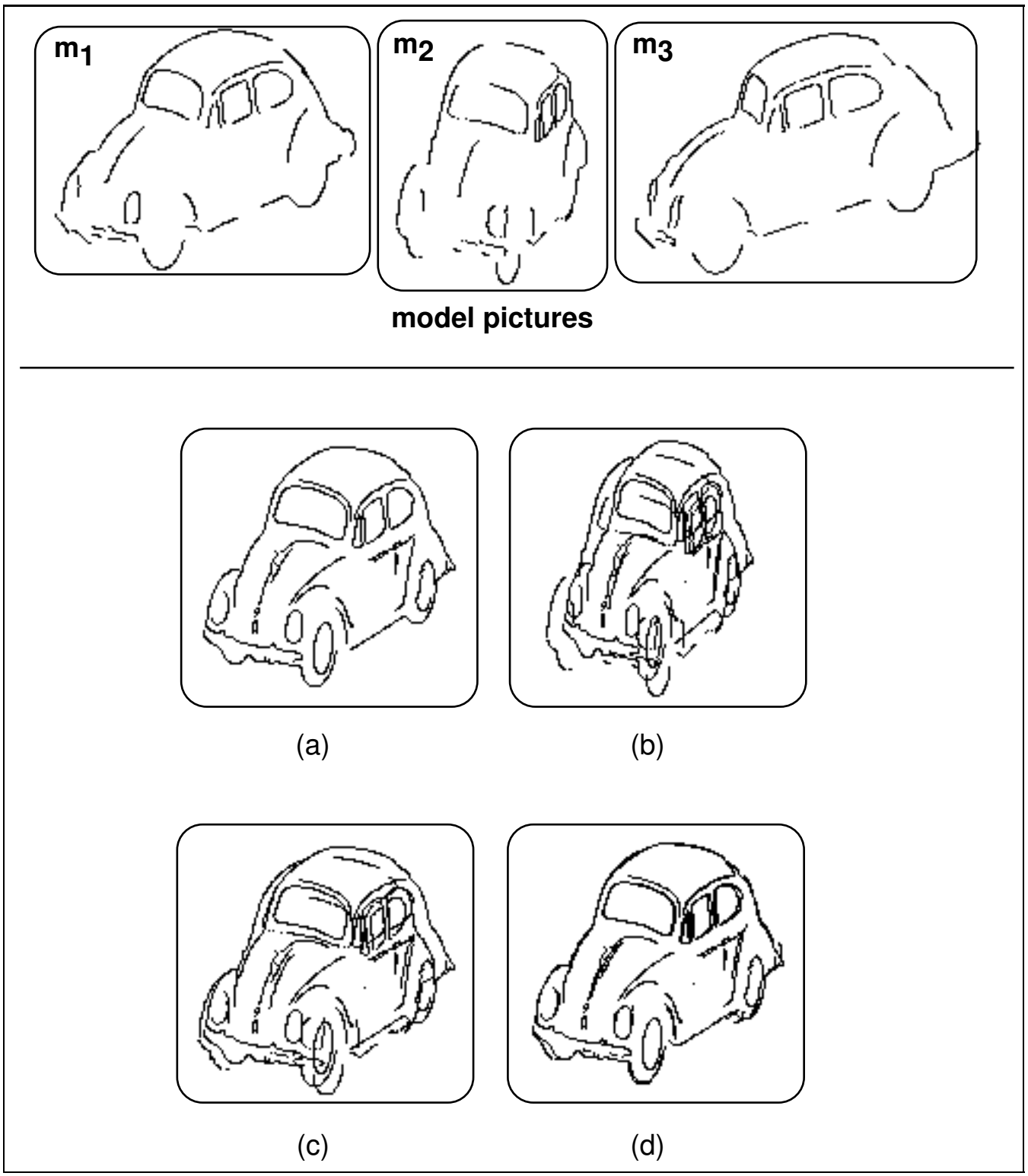

Figure 12: *Example of our recognition process on a toy VW and a real complex novel image.(a) The novel image of the VW. (b) We used the second model picture m_2 in the matching process. A comparison of the roughly aligned novel image to this model picture is shown. The result sections contains (c) the transformed model superimposed on the novel image after one iteration and (d) the transformed model superimposed on the novel image at the end of 13 iterations.*

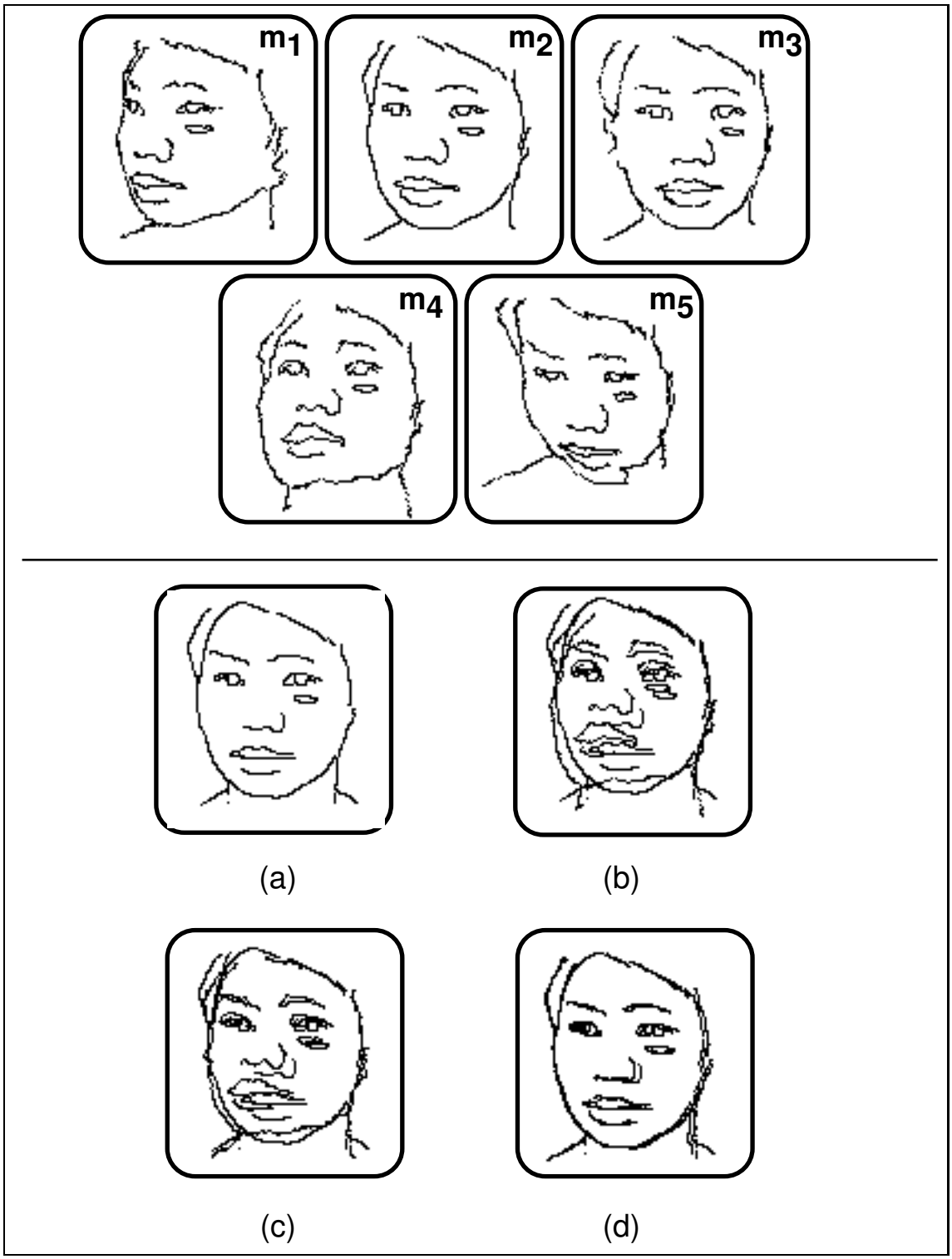

Figure 13: *Example of the correspondence process on a human face (KKS). The five model pictures of the face are shown above the horizontal line. (a) A real novel image of KKS whose face pose is between the face poses in model pictures 2 and 3. (b) The novel image is compared to model image 4. A superimposition of the two prior to the matching process is shown here. The result of the matching process (c) after 1 iteration and (d) after 10 iterations; the transformed model is superimposed on the image.*

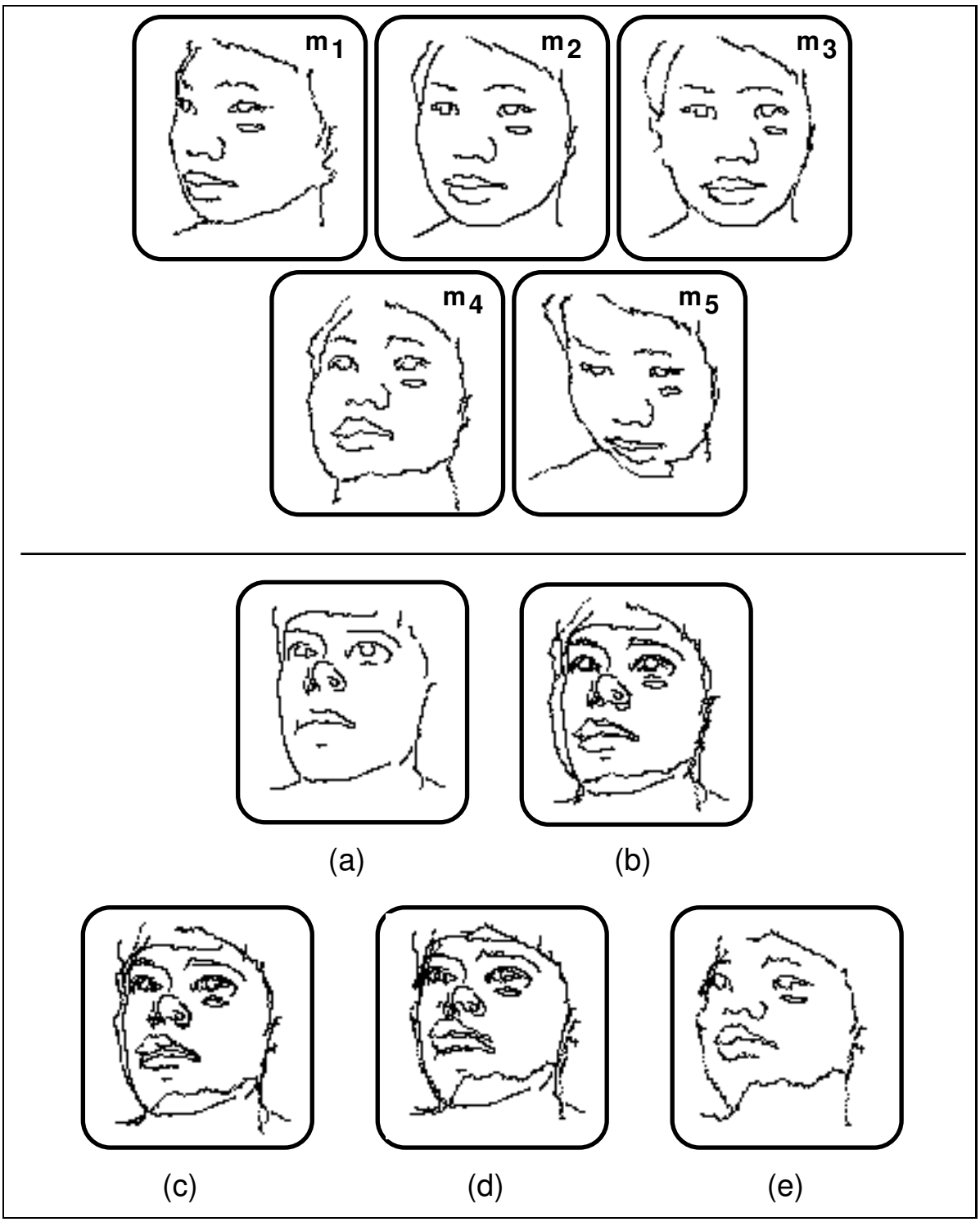

Figure 14: *Example of the correspondence process acting on a model of a face of KKS and a novel image from a different person (DB). (a) The novel image of DB. (b) A comparison of DB to KKS model image 4 prior to the matching process. (c) The result of matching the novel image to KKS model image 4 after one iteration; the transformed model is superimposed on the image. The result of the correspondence process after 10 iterations; (d) the transformed model is superimposed on the image and (e) the transformed model is shown alone*

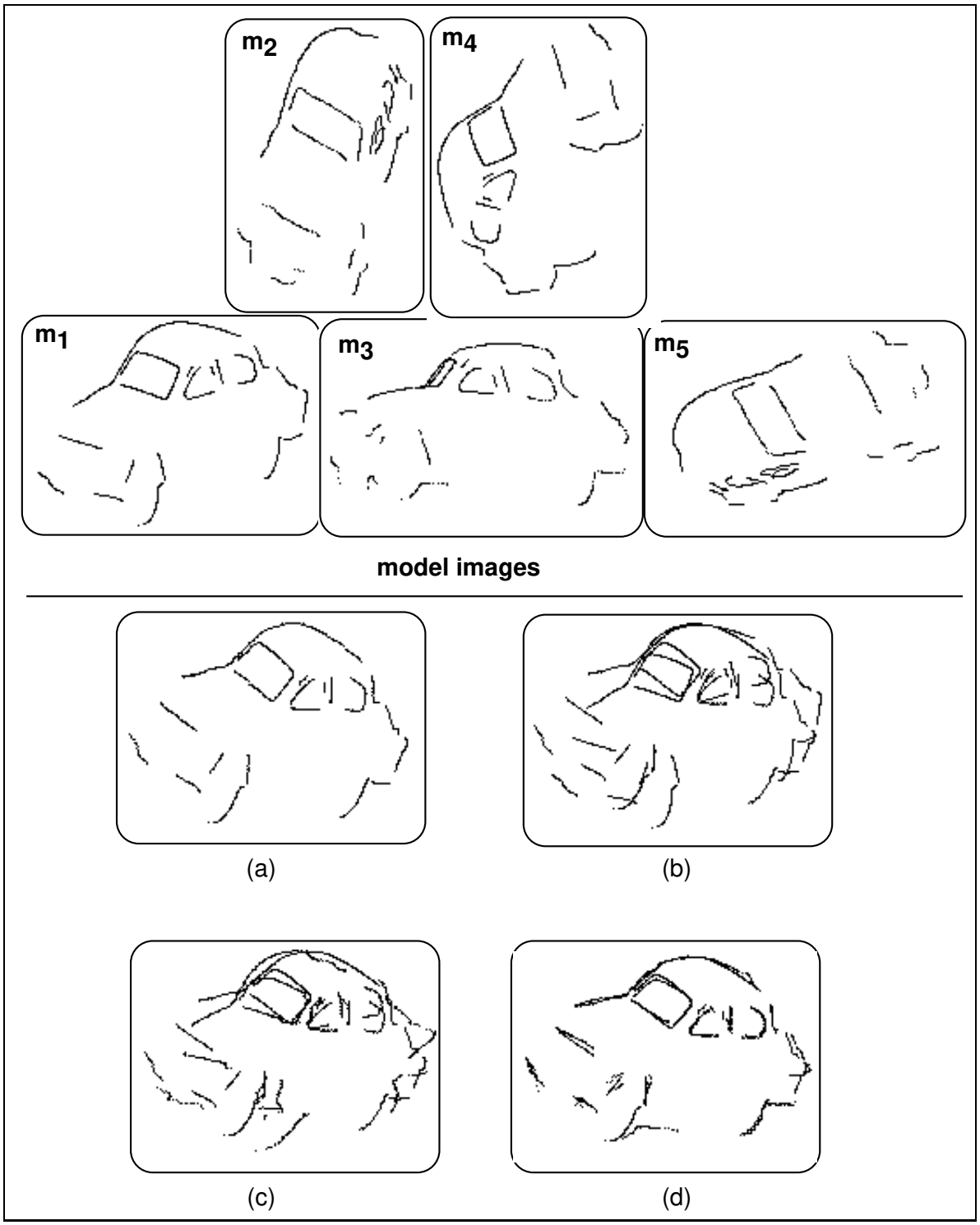

Figure 15: *Experiment using incorrect compensation for rotation in the xy plane. Five model pictures of a toy Saab are shown above the horizontal line. (a) The novel image was created by rotating model picture m_1 by 18 degrees. (b) The correspondence process initially compares the novel image to model picture m_1 without any compensatory adjustment. The results of the process are shown in (c) the transformed model superimposed on the image after the first iteration and in (d) the final result; a comparison of the transformed model and the novel image after 13 iterations*

References

[1] I. Bachelder. Contour Matching Using Local Affine Transformations. Master's thesis, Massachusetts Institute of Technology, June 1991.

[2] D.H. Ballard. Generalizing the Hough Transform. *Pattern Recognition*, vol. 13, no.2, pgs 111-122, 1981.

[3] H.H. Bulthoff and S. Edelman. Psychophysical Support for a Two-Dimensional View Interpolation Theory of Object Recognition. *Proceedings of the National Academy of Science*, Vol. 89, pages 60-64, January 1992.

[4] T. Cass. Feature matching for object localization in the presence of uncertainty. A.I. Memo 1133, Artificial Intelligence Lab, MIT, 1990.

[5] S. Edelman and H.H. Bulthoff. Viewpoint-Specific Representations in Three Dimensional Object Recognition. A.I. Memo 1239, The Artificial Intelligence Lab, MIT, 1990.

[6] S. Edelman and T. Poggio. Bringing the Grandmother Back into the Picture: A Memory- Based View of Object Recognition. A.I. Memo 1181, The Artificial Intelligence Lab, M.I.T., 1990.

[7] W.E.L. Grimson and D.P. Huttenlocher. On the sensitivity of the hough transform for object recognition. A.I. Memo 1044, The Artificial Intelligence Lab, M.I.T, 1990.

[8] W.E.L. Grimson. On the recognition of curved objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):632-643, 1989.

[9] W.E.L. Grimson and D.P. Huttenlocher. On the verification of hypothesized matches in model-based recognition. A.I. Memo 11110, The Artificial Intelligence Lab., M.I.T., 1989.

[10] W.E.L. Grimson and T. Lozano-Perez. Model-based recognition and localization from sparse range or tactile data. *The International Journal of Robotics Research*, 3(3):3-35,1984.

[11] E.C. Hildreth. The Measurement of Visual Motion. The MIT Press, Cambridge, 1984.

[12] E.C. Hildreth. The neural computation of the velocity field. In *Vision and the Brain*, pages 139-164, 1990.

[13] B.K.P. Horn and B.G. Schunk. Determining optical flow. *Artif. Intell.* 17:185-203, 1981.

[14] B.K.P. Horn. Robot Vision. The MIT Press. Cambridge, MA. 1986

[15] D.P. Huttenlocher and S. Ullman. Object recognition using alignment. *International Conference on Computer vision*, pages 102-111,1987.

[16] P. Lipson. Model Guided Correspondence. Master's thesis, Massachusetts Institute of Technology, June 1993.

[17] D.G. Lowe. Perceptual organization and visual recognition. Technical Report STAN-CS-84-1020, Stanford University, 1984.

[18] D.Marr and S.Ullman. Directional selectivity and its use in early visual processing. *Proc. R. Soc. London Ser.*, B211:151-180.

[19] S.B. Pollard, J. Porrill, J.E.W. Mayhew, and J.P. Frisby. Matching geometrical descriptions in three-space. *Image and Vision Computing*,5(2):73-78, May 1987.

[20] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. Numerical Recipes in C. Cambridge University Press. Cambridge, MA. 1988.

[21] I. Rock. The Logic of Perception. M.I.T. Press, Cambridge, MA, 1983.

[22] A. Rosenfeld and A. Kak. Digital Picture Processing. Vol. 2. Academic Press, Inc. San Diego, CA, 1982.

[23] D. Shoham and S. Ullman. Aligning a model to an image using minimal information. IEEE, pages 259-263, 1988.

[24] G. Strang. Introduction to Applied Mathematics. Wellesley-Cambridge Press. Wellesley, MA. 1986.

[25] S. Ullman. Aligning pictorial descriptions: An approach to object recognition. *Cognition*, 32(3):193-254, August 1989.

[26] S. Ullman and R. Basri. Recognition by linear combinations of models. A.I. Memo 1152, The Artificial Intelligence Lab., M.I.T., 1989.

[27] P. Viola and W. Wells. Alignment by Maximization of Mutual Information. *Fifth Internation Conference on Computer Vision*, pages 16-23, June, 1995.