MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

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

Model Based Correspondence for Object Recognition

Pamela Lipson and Shimon Ullman

Abstract

We present a new technique to register an object model with an observed image. Our 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 process in order to reduce the inherent complexity of the problem and to increase the robustness and efficiency of the solution. We begin by roughly aligning image and model features. We then determine an estimate of a correspondences between a sparse set of model and image contours. Finally, we constrain the rest of the matches via global information from the model. These stages are 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 computationally simple and efficient; the use of models effectively constrains the matches and the process results in a linear solution. Second, it is effective; experiments show that our procedure quickly converges to a solution, if one exists. Finally, the procedure is robust; moderate errors in the rough alignment stage do not impair the subsequent correspondence procedure.

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 [20][Bruel, Todd]. 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 [14, 16, 6, 12]. Feature correspondences are computed in a "bottom up"manner, where image features, mainly pointwise, are matched with model features. Transformations based on these combinations are computed and the result is evaluated. To reduce complexity of the possible feature correspondences, such 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 the model or "top down" information to determine the feature correspondences.

Our motivation for using model-based information in the pairing process, lies in two problems which plague "bottom up" approaches. First, the abstraction process of translating image attributes into more abstract features is performed without the use of any model/scene specific knowledge. As a result, the chosen image features may not have much meaning with respect to the features present in the model [2]. In addition, the heuristic rules to eliminate incorrect matchings are often ineffective because local image feature and model feature properties are usually insufficient to rule out possible pairings (see figure 1).

We suggest that a more efficient and robust correspondence technique can be created by introducing "top down" or model-specific knowledge in the matching process. This type of approach would allow model-based knowledge to influence the extraction of features from the input images and to constrain the set of possible matches. (In essence, this allows model information to be compared to image information at any level in the flow of computation.)

The use of model-driven data has three main advantages. First, such an approach offers a great latitude in the types of features to match. Model and image features may no longer be constrained to non-distinctive constructions such as points, T-junctions, and corners nor to non-robust constructions such as complex blobs. Constraints from the model may allow us to match robust *and* distinctive (but hard to characterize) features such as curves. A second advantage is that we can roughly distort the model to match the image. This has the advantage of bringing model features into closer spatial correspondence thereby shrinking the search space

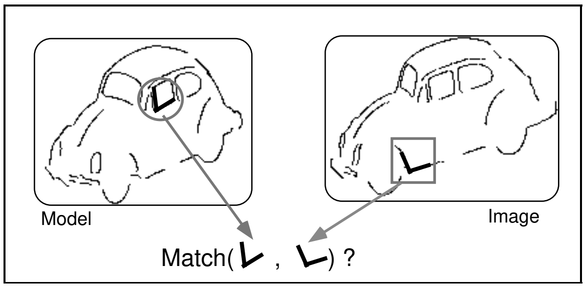

Figure 1: *One of the disadvantages of using only "bottom up" information in the matching process. Two pictures of a VW are shown. One represents the model and the other the image. The goal is to find a correspondence between image and model features. One model corner feature and one image corner feature 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.*

and reducing the number of false matches. The third advantage is that we can choose the set of model and image correspondences that are globally consistent with the model. This added constraint further reduces the problem's complexity (see figure 2).

In this paper, we present a solution to the registration problem that utilizes model driven information to determine corresponding model and image features. The remainder of this paper presents our model-guided approach to registration within the framework of a model-based recognition system. The particular system we implemented uses view-based models, contours as features, and a linear combination recognition engine. Sections 2 and 3 discuss the motivation behind our choice of pictorial models and contour features. Section 4 argues that our approach to correspondence is especially effective in these domains. In section 5, we present the details of our model-guided contour matching algorithm. Section 6 describes our implementation. In section 7, we show examples of our approach tested on a range of synthetic and natural models and images within a recognition application.

2 View-based models

The choice of object model representations is critical to the correspondence stage. The chosen representation influences the type of model features used, the type of image features extracted, and the internals of the matching process. Models can be either two-dimensional or three-dimensional. They can range in complexity from sparse (i.e. small point or edge sets) to detailed representations (i.e. sets of complete images).

It would seem that the obvious way to encode a three-dimensional object is via a three-dimensional model. Such a representation is complete in that it implicitly contains full information about all the views of the object, the connectivity of subparts, and depth information. The disadvantages of this model type include the difficulty involved in generating and storing the detailed model and determining a transformation of the model to

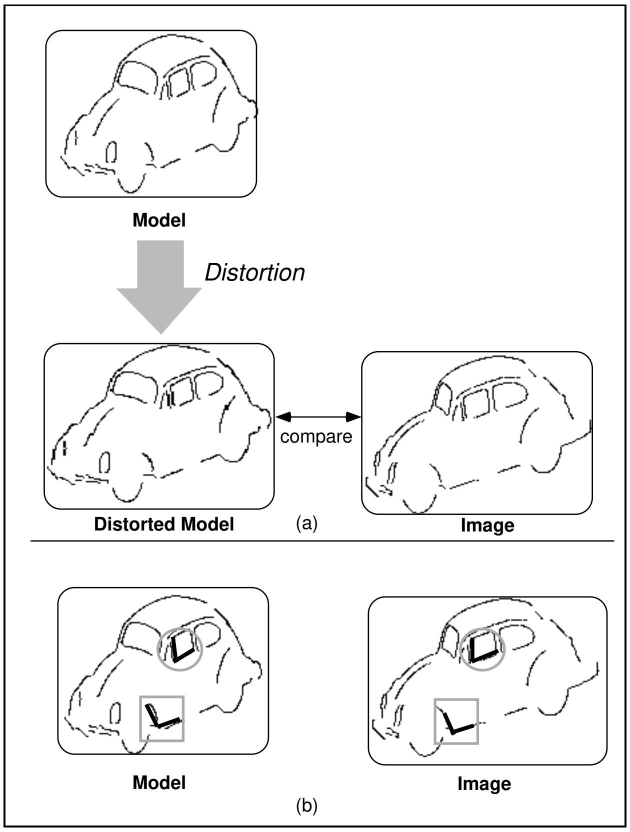

Figure 2: *The advantages of adding "top down" information to the matching process. The two pictures of the VW are replicated from figure 1. (a) Using "top down" information, we can distort the model to match some characteristics of the image in order to bring model and image features into closer spatial proximity. (b) We can choose image and model feature pairs that are globally consistent with the model. This figure shows two corner features highlighted on both the image and the model. Unlike the ambiguity we found in figure 1, our model-based constraints suggest we reject the match of the model feature in the circle and image figure in the square for the more globally consistent match of model and image feature in the circles and the match of the model and image figures enclosed in squares. Both (a) and (b) shrink the space of possible matches and reduce the possibility of false correspondences.*

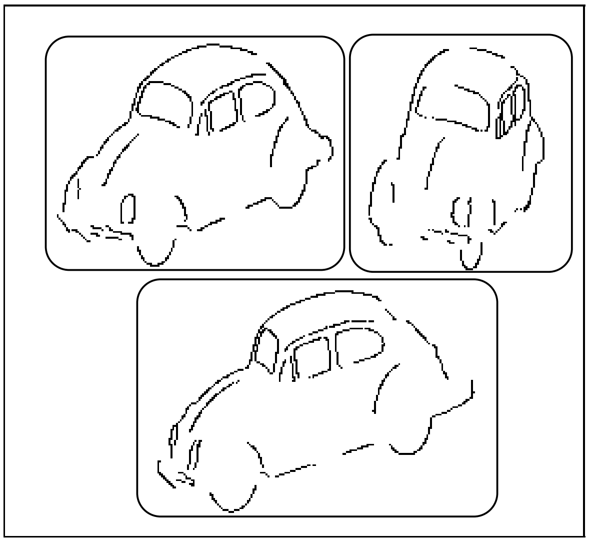

Figure 3: *Example of a pictorial model of a VW.*

2D views so that it can be compared with the observed two-dimensional image. In addition, recent psychophysical evidence has shifted the attention away from using full three-dimensional models [3, 4, 5]. These studies suggest that the human visual system recognizes images by matching them with previously stored two-dimensional views without the use of three-dimensional representations.

Given the computational disadvantages of 3D models and the psychophysical evidence against them, some recent approaches have attempted to construct models of 3D objects using sets of two-dimensional pictures (see figure 3). This modeling strategy is known as the pictorial approach. One scheme which uses pictorial models is recognition by linear combination of views [19]. The linear combination approach utilizes the fact that a picture of an object decomposed into points can be represented by a linear combination of the corresponding points on a fixed number of other 2D views of the same object. The benefits of incorporating pictorial models into a model-based recognition scheme are that 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.

3 Using contours as image and model features

A good feature representation should be distinctive, reliable in the face of noise and occlusion, stable over changes in viewpoint, easy to compute, and easily encodable. Unfortunately, no single feature type explored thus far has been able to satisfy simultaneously these criteria. Features that are distinctive such as object parts are often prone to noise and occlusion, unstable over changes in viewpoint, difficult to extract from images, and expensive in terms of storage. Alternatively, those that are easy to compute and robust such as points or vertices are often indistinguishable from each other. A

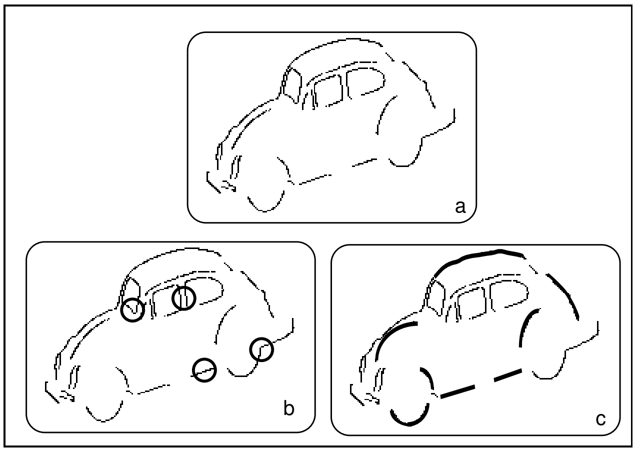

Figure 4: *Representations for Features. (a) A real image. (b) Some possible explicit features such as points, lines, corners, and parallel lines. (c) A few contours that could be used as features are highlighted.*

poor feature representation can greatly hinder the success of any correspondence process.

Figure 4 shows a line representation of a car. 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 [18]. Some of these explicit features are highlighted in figure 4b.

Looking again at the explicit features in 4b, 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. In addition, they can be viewed as local features and also seen as a global arrangement, providing a solution to the distinctiveness vs. robustness problem. They also allow for some invariance to lighting and changes in viewpoint. The use of contours can, therefore, be expected to facilitate the correspondence process.

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. There is an inherent ambiguity in matching contours. Additionally, they are hard to characterize, and therefore, difficult to use as "explicit" features. Both issues must be addressed before they can be used as features in a model-based recognition system.

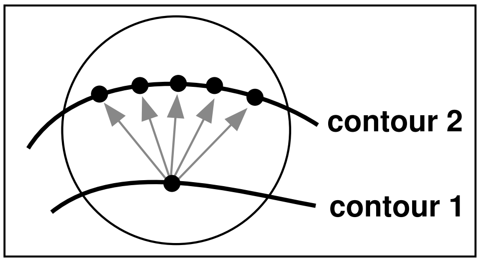

Figure 5: *The aperture problem - Looking through a local aperture a point on contour 1 may match with any of an infinite number of points on contour 2.*

4 The use of "Top down" information for contour matching

The idea of general contour matching is best formulated in work related to the recovery of optical flow fields from short-range motion. The basic idea is to best recover the structure of a moving three-dimensional object from a series of two-dimensional pictures, taken over short temporal and spatial intervals [9, 11]. This is performed by tracking a set of features in each of the frames. The major problem in computing optical flow fields is that only motion perpendicular to these features can be accurately established [15]. This is commonly known as the aperture problem (see figure 5).

Because contour matching is ambiguous, attempts to resolve the problem within the optical flow domain have been based on general constraints such as smoothness or assumptions of local affine motion fields [1, 10]. One could utilize the ideas developed in the field of apparent motion to match contours in a recognition application.

In the domain of object recognition we actually have additional information to aid the matching process. Here we are dealing, by definition, with familiar objects and therefore can use the knowledge associated with these objects to try to overcome the aperture problem.

The downside is that within a recognition domain we have no time varying imagery to work with. Thus, we cannot technically compute the normal component of motion required in the optical flow schemes described above. In addition, different views of an object often appear highly dissimilar, violating the close spatial proximity assumption of short-range motion.

"Bottom up" correspondence strategies do not utilize any of the benefits derived from model information. Without any global information there is no meaningful way to interpret the image as a global arrangement of local contours. In addition, "bottom up" strategies have no way to overcome the aperture problem. Therefore, they are insufficient to cope with the disadvantages of characterizing and matching contours.

We suggest that the addition of "top down" or model information provides useful constraints for the interpretation of the image data. 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. The model

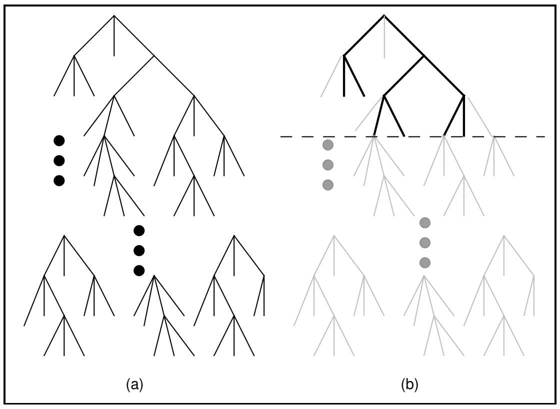

Figure 6: *The advantages of using the model to guide the correspondence process over a purely "bottom-up" approach. (a) The "bottom" up approach may have to evaluate all the possible pairings of image and model features. The graphic depicts a search tree where every model feature is compared to every 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.*

also provides the information necessary to compute an analog of the "normal component" of motion. Finally, given this model-based "normal component of motion" one can incorporate techniques related to motion analysis to determine a fine correspondence between contours.

5 Our scheme for correspondence

In our work we attempt to demonstrate that the model ("top down data"), in conjunction with "bottom up" data, can be used toguide the correspondence process. 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 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 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 6.

Our proposed algorithm for computing correspondence between a model and an image is summarized pictorially in figure 7. The inputs to the system are a set of two-dimensional model pictures and an image. We use 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 im-

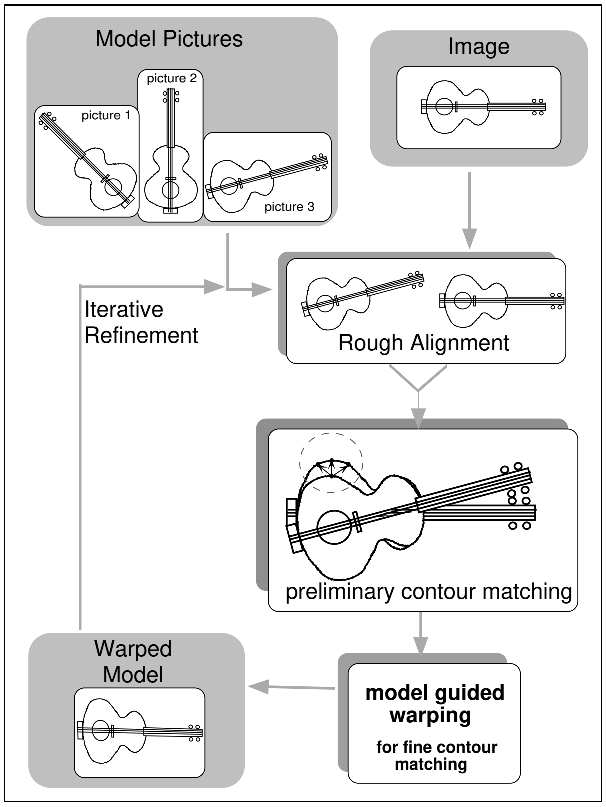

Figure 7: *Summary of the proposed correspondence process. The process begins by roughly aligning the image with a model picture. We then postulate rough matches between a few model and image contour points in a "bottom up" fashion. These rough matches are sent to the next stage where they are refined 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.*

age 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. *Sparse "Bottom-up" Correspondences.* 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.* 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. The algorithm uses the idea on our result that the information from the model allows contour matching to become possible within a recognition application. 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 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.

Our theoretical and experimental results demonstrate that our technique is efficient, proceeding in a linear fashion with few iterations, and robust, allowing for significant errors in the rough alignment stage. In the next section, we describe the details of our correspondence algorithm.

6 Our Implementation

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.

We have chosen to test our model-driven approach to correspondence by embedding it in the linear combination of views framework for object recognition [19].

6.1 The Linear Combination Approach to Recognition

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, $\mathbf{m_1}, \mathbf{m_2}, \ldots \mathbf{m_n}$, 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 [19].

We now describe our proposal for solving the correspondence problem. As mentioned in section 5 it has four parts: rough alignment, sparse "bottom up" correspondences, model-guided matching, and verification and iterative refinement.

6.2 Rough Alignment

Our correspondence process begins by trying to partially compensate for gross changes in the viewing geometry between an image and a model using only minimal information [17]. The rough alignment involves an affine transformation of the image (or model) to compensate for differences in in-plane translations, in-plane rotations, in-depth displacements, small in-depth rotations, and small non-rigid transformations [18].

We utilize a 4 stage rough alignment procedure. In general, the compensatory rough alignment transformations must be robust in the face of noise and especially occlusion. Although a complete solution to this problem is beyond the scope of this paper, we do suggest some simple procedures to determine a coarse registration of object and model which we have found to be effective in our experiments. We illustrate our proposals with the model and image picture pair shown in the first row of figure 8.

1. *Compensating for translations in xy plane.* Looking at the top row in figure 8, the figure of the guitar has been translated to the upper left corner of the image with respect to the model picture. We can compensate for

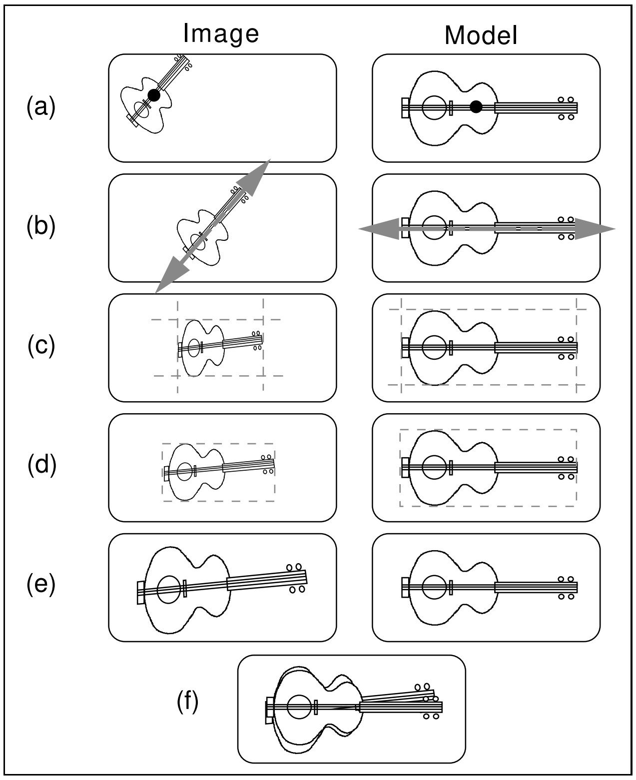

Figure 8: *The four compensatory transformations involved in our rough alignment process. (a) Aligning the centers of mass compensates for translation in the x-y plane. (b) Aligning the dominant orientations compensates for rotations in the x-y plane. (c) Independent scaling compensates for small rotations in depth, while (d) uniform scaling compensates for translations in depth. (e) shows the net result after all the compensations. (f) After the rough alignment stage, the model and image should be in approximate registration.*

this translation in the xy plane by aligning the centers of mass of the figure in the image and in the model picture. The centers of mass can be determined by calculating the first moments of the image and model points.

2. *Compensating for rotations in xy plane.* One method to account for in-plane rotations is to align the dominant orientations of the figures in the image and model pictures. 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. *Compensating for rotations in depth.* Small rotations in depth of a 3D object cause its 2D projection to appear as if it has been scaled independently in the x and y directions. This is illustrated in the third row of figure 8. We can measure the overall and dimensions of the figures in the image and model picture. Such a procedure is analogous to determining a bounding box for the figures. Scaling independently in the x and y directions can compensate for small in-depth rotations or shearing transformations.

4. *Compensating for translation in depth.* Translations in depth of a 3D object cause its 2D projection to appear as if it has been scaled uniformly in the x and y directions. 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.) We can scale the image figure uniformly in both dimensions to compensate for this difference.

After transforming the image based on compensations for translation, rotation, shearing, and scale we can compare the new image to the model. The result is a good but not perfect alignment of the image and model picture. The subsequent correspondence stages are designed to refine this alignment.

6.3 "Bottom Up" Matching of Sparse Features

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. We now approach the matching of contours in a "bottom up" fashion. The output of this stage is a set of possible matching points between a few 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 an arbitrary sampling of all the model points. Based only on "bottom up" information of proximity, we attempt to match each chosen model contour point m with its corresponding point on an image contour. For each of these points m, using a radial

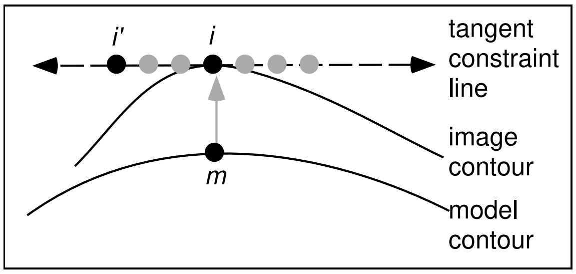

Figure 9: *We represent contours by local tangent lines. Our assumption is that the match to a model contour point, m, is either the radially closest image point, i, or a point, i′, along the contour-tangent at i.*

search, we produce a guess 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 somewhere close i along the image contour containing i.

We represent contours as a collection of local tangent lines. We use the tangent constraint to estimate the matches between image and model contour points. Our tangent representation for contours simplifies the problem of choosing a point along a non-parameterized contour to choosing a point along a parameterized line. We now assume that the corresponding point to m is either point i or some point which lies on the tangent to the image curve at i (see figure 9). It is the goal of the subsequent fine alignment stage to condense the guess from all the points on the tangent to a tangent point i' that is closest to the true corresponding image point.

We argue that the local tangent of the contour is a sufficient approximation to the curve. Although the final decision may not be a point lying on the image contour, we have designed the procedure such that we will eventually 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. Our tangential assumption states that

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

In equation 2, the known or derivable variables are (i_x, i_y) and (v_x, v_y). (i_x, i_y) is determined from "bottom-up" inspection of the model and image data. (v_x, v_y), which is the local orientation of the tangent to the curve at i, can be computed using second moment analysis. We can remove t from the equation by uncoupling the x and y components of equation 2 and solving for t. The result is the equality

$$v_y i'_x - v_x i'_y = v_y i_x - v_x i_y \qquad (3)$$

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.

6.4 Model-guided Matching

The model-guided matching stage 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 10).

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 we can write the equations

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

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

We will use this set of constraints to recover the coordinates for i' (which, as described in the last section is the matching point in the image for the given model point).

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

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

Now the only unknowns are the linear combination coefficients $(\alpha_{1x}, \alpha_{2x}, ..., \alpha_{nx})$ and $(\alpha_{1y}, \alpha_{2y}, ..., \alpha_{ny})$. We have, thus, reduced the correspondence problem to solving for these coefficients such that they satisfy equation 6. Once we recover these coefficients, we can use equations 4 and 5 to determine the coordinates of i', which is our ultimate goal.

Our technique to determine $(\alpha_{1x}, \alpha_{2x}, ..., \alpha_{nx})$ 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 $\mathbf{m}$. As a result, we can construct an overdetermined set of linear equations.

The following is equation 6 in matrix form.

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

We can use several methods to solve for these coefficents based on the linear equations. One technique is

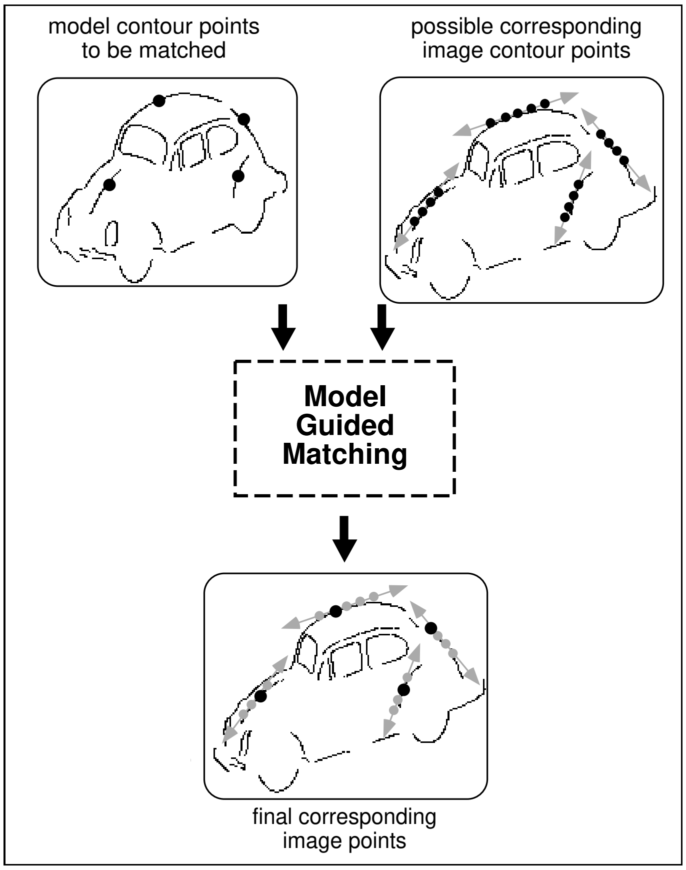

Figure 10: *The fine alignment stage takes as input the set of possible image contour points that match the selected model contour points. It then uses model guided constraints to pick a unique correspondence for each model point.*

to use pseudo-inverse to calculate the best average solution for the unknowns (see [13] for more details). An alternative solution is to use a voting technique, such as the hough transform, which would provide us with most consistent coefficients for the majority of the equations.

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

6.5 Verification and Iterative Refinement

The coefficients recovered in the model-guided matching stage represent a *hypothesis* as to the transformation that takes the model onto observed image. We can *verify* our hypothesis by computing the linear combination of the model pictures and then comparing the result with the observed image. If we have some estimate of tolerance for error in the mapping, we can determine if our solution is correct.

One of the benefits of our 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 chosen model picture . It can be compared to the image using the same techniques described in sections 6.2, 6.3, and 6.4 (see figure 11). If the image is truly an instance of the model, repeated refinements of the coefficients would converge to an improved solution.

6.6 Analysis

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 12a depicts the scenario where contour points from the model are incorrectly matched to contour points in an image because of an uncorrected large displacement. Experimentally, we found that the effect on the process by these incorrect matches can be partially compensated for by correct contour matches at other places in the model and image. Through a series of iterations of the correspondence process, 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 usefule to incorporate more information into the contour correspondence process. One possibility is to look for geometric similarities between contours as a matching criteria. Another is to use information about whether contours are labeled as interior or boundary curves.

Figure 12b 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

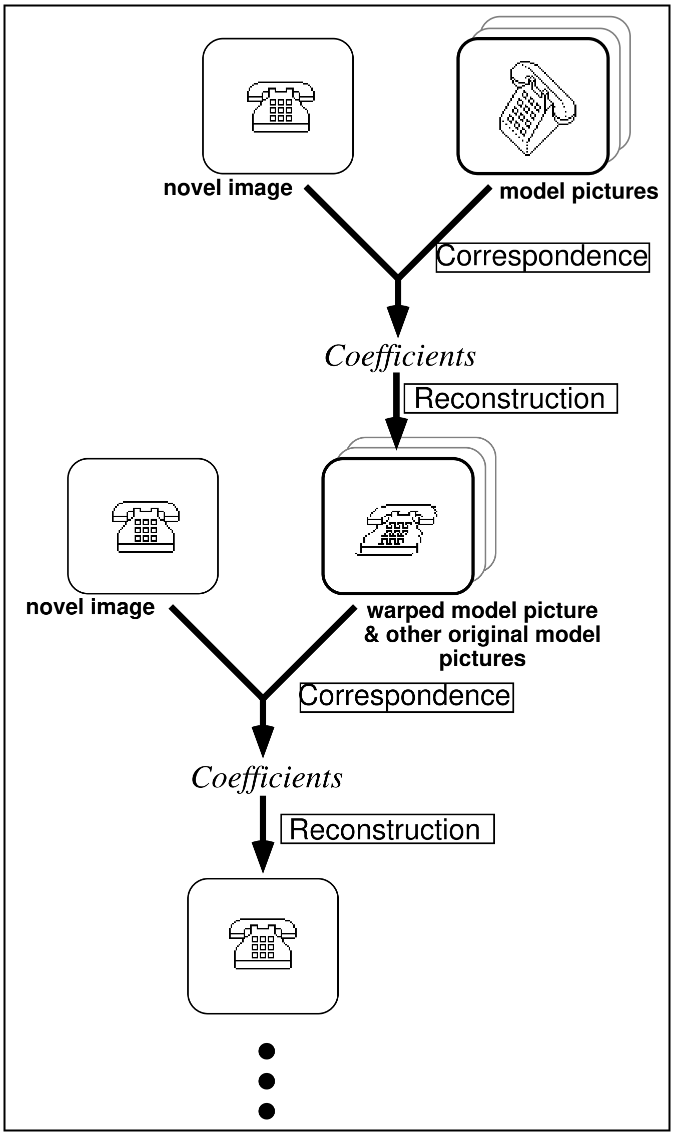

Figure 11: *The process of iterative refinement. After we compute the linear combination coefficients by matching one of the original model pictures and the novel image, we can transform the model and compare the result to the observed image. If we desire a refinement of the coefficients, the transformed model picture can take the place of one of the original model images. We can then repeat the correspondence process with this modified set of model pictures.*

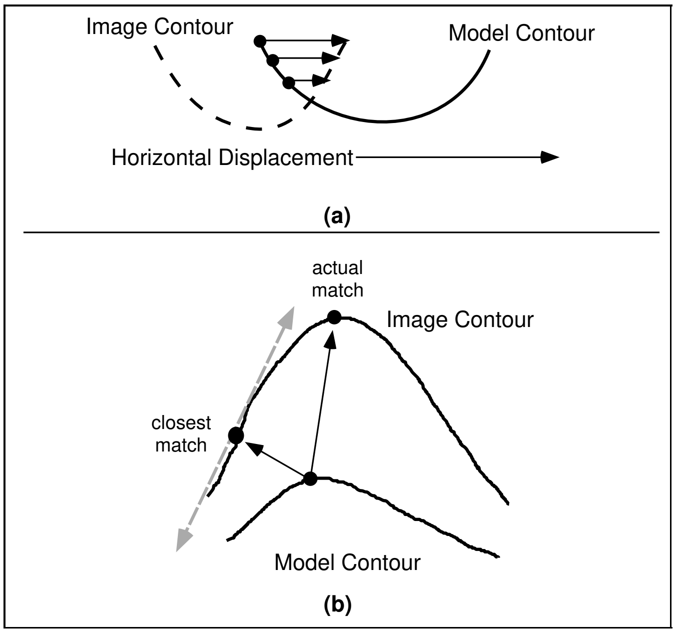

Figure 12: *Two potential sources of error in the "bottom up" feature matching stage: (a) errors due to large displacements between the image and model contours (b) and errors due to high curvature contours.*

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.

7 Results

The results of our approach for computing correspondence can be evaluated in terms of accuracy and speed. We implemented our method in C on a Sun Sparc IPC workstation. We tested our procedure on two-dimensional polygons, two-dimensional closed curves, synthetic three dimensional polyhedra, and real imagery. All images were pixel 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 [19] for an explanation of the number of images that makeup a model).

The model pictures each contained between 1000-1500 contour points. Typically, the number of points used in the sparse bottom up matching phase (see section 6.3) ranged from 50 points for the 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 tried to compute correspondences between a model and a novel image. For a given model, we tested our procedure on novel images that were instances that model and novel 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.

7.1 Results with Accurate Rough Alignment

In this section, we report the results of our experiments on synthetic polyhedra and real imagery. In each of these examples, the rough alignment stage was able to bring the chosen model image and input image into a good registration.

7.1.1 Synthetic Polyhedra

Figure 13 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. We added a fourth picture, containing points only at (1,1), to account for any translational component not compensated for in the rough alignment stage. The pose of the novel image is half way between the pose of the second model picture ($\mathbf{m_2}$) and and third model picture ($\mathbf{m_3}$). To fully test the system, we used the most dissimilar model picture to the image, model picture 1 ($\mathbf{m_1}$), to compute the contour matches. Figure 13a shows the novel image. Figure 13b shows the image and model picture 1 roughly aligned and superimposed before the start of the matching process. Figure 13c contains the result after the first contour matching iteration. Here the transformed model is superimposed on the image. Finally, Figure 13d shows the transformed model superimposed on the image after 3 iterations of the correspondence process. The transformed model and image are visually indistinguishable. (A sum of squares error analysis show that they differ on average by 0.4 pixels/ model contour point)

7.1.2 Real Imagery

We used real pictures of a miniature Volkswagen, a miniature Saab, and a human face to further test our correspondence technique. We divided the testing into parts. The first part restricts the possible transformations of the objects. Here we specified that only rotations around one axis could be used to generate the model pictures and images. The second part utilizes full 3D rotations. Figures 14 and 15 are examples of the ability of our system to compute correspondences with real imagery. With the computed correspondences, our system was able to recognize new images of the cars and faces using only a few prestored pictures.

In figure 14, a toy VW was rotated around the y axis to create three model pictures. The image, shown in figure 14a, is derived from a linear combination of the three model pictures. The recipe for this particular image is $0.5\mathbf{m_1} + 0.5\mathbf{m_2}$. We compute feature correspondences between the first model picture ($\mathbf{m_1}$) and the image. In figure 14b, we show the differences between the image and model picture 1. Figure 14c shows transformed model superimposed on the image after 1 iteration. The final picture illustrates the effectiveness of our correspondence technique by superimposing the final transformed model, derived from 10 iterations of the matching process, onto the novel image. The two are almost visually indistinguishable.

The five model pictures in figure 15 show a human face (KKS) under various three-dimensional rotations. The novel image shown in 15a is a new image of the face whose pose lies between the pose of the face in model pictures 2 and 3 ($\mathbf{m_2}$ and $\mathbf{m_3}$). For the first iteration of the correspondence process, we compare the image to model picture 4 ($\mathbf{m_4}$). We specifically chose this model picture for the matching because it was quite dissimilar to the image. Figure 15b shows the comparison of the image to model picture 4. Figure 15c shows the result of the first iteration of correspondence process. Figure 15d 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.

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

We also compared a novel image of a different face (DB) to our original face model (KKS). Figure 16 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 16e shows the transformed model after 10 iterations of the matching process. The average sum of squares displacement 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 our face model. In these cases, the cars had few features in common with the face model and the iteration process quickly diverged.

7.2 Results with Inaccurate Rough Alignment

The assumption of accurate rough alignment may not be valid in many real world cases. 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 remains a problem. The subsequent example uses an image and a model picture that differ by a rotation in the $x-y$ plane. For the most part, the system could tolerate rotations of at most 24 degrees. After 24 degrees, the majority of the image and model contours violated our spatial proximity assumption. As a result, there was no evidence of convergence despite repeated refinements. Below the 24 degree mark the system required at approximately 10 additional refinements before a converging to a final solution.

Figure 17 shows a Saab model created from various three-dimensional rotations of the car In this case, the image is an inplane rotation of model picture 1 ($\mathbf{m_1}$) by 18 degrees. The inputs to the correspondence process are model picture 1 and the image without any inplane rotation compensation. Figures 17c and 17d show a comparison of the transformed model and the image after first and 10th iterations respectively. By the tenth iteration, the process has converged to a reasonably good solution.

7.3 Results Using Only Pointwise Matches

We repeated all of our experiments without the model-guided contour matching stage detailed in section 6.4. The modified process consisted of a rough alignment stage followed by matching model contour points to their closest image counterparts. We used these pointwise matches to solve for the linear combination coefficients. The modified process was then repeated to refine the model to image correspondence.

We found that under the conditions of accurate rough alignment and a good distribution of model points, this modified technique, at best, took twice as long to converge than our original technique. Under unfavorable conditions, inaccurate rough alignment and a poor distribution of model points, the modified technique often failed to converge.

8 Conclusion

Our goal was to create an robust and efficient technique which could compute a model to image correspondence. We have proposed a scheme to achieve this by blending both "bottom up" and "top down" information. There are three benefits to our approach. First, it is computationally simple and efficient; the use of models constrains the matches and allows for a linear solution. Second, it is is effective; experimentally we have found that iterative refinement of the process quickly converges to a good solution, if one exists. Finally, the procedure is robust; even significant errors in the rough alignment stage do not impair the subsequent correspondence procedure.

We can enhance the behavior of the performance of the system as a whole by further developing the rough alignment stage. In section 6.2, we have presented some simple techniques to compensate for differences in viewing geometry used to generate the model picture and the image. These techniques for rough alignment, however do not sufficiently handle large in-depth rotations, objects without a dominant orientation, non-rigid objects, or occlusion. If the rough alignment procedure were more robust, the subsequent correspondence stage would benefit greatly. First, we could perform fewer "bottom up" contour matches. Second, the need for repeated refinements of the correspondence would be reduced. Also, an improved rough alignment procedure would allow the technique to use model views which coarsely sample the view space.

We also desire a more sophisticated verification criteria. Once we transform the model, we compare the resulting picture to the image using the sum of the squared displacements between corresponding model and image points. This measure is inadequate to distinguish between a non- rigid transformation of the model and a completely new object.

In the future, we would plan to incorporate our procedure for correspondence into other model-based recognition systems. We are especially interested in testing our technique in conjunction with recognition using view approximation by regularization networks and recognition by linear mapping. Since these both require exact point to point matches, we feel our correspondence algorithm will prove to be of use to them..

Acknowledgments

The authors would like to thank Ronen Basri, David Beymer, 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.

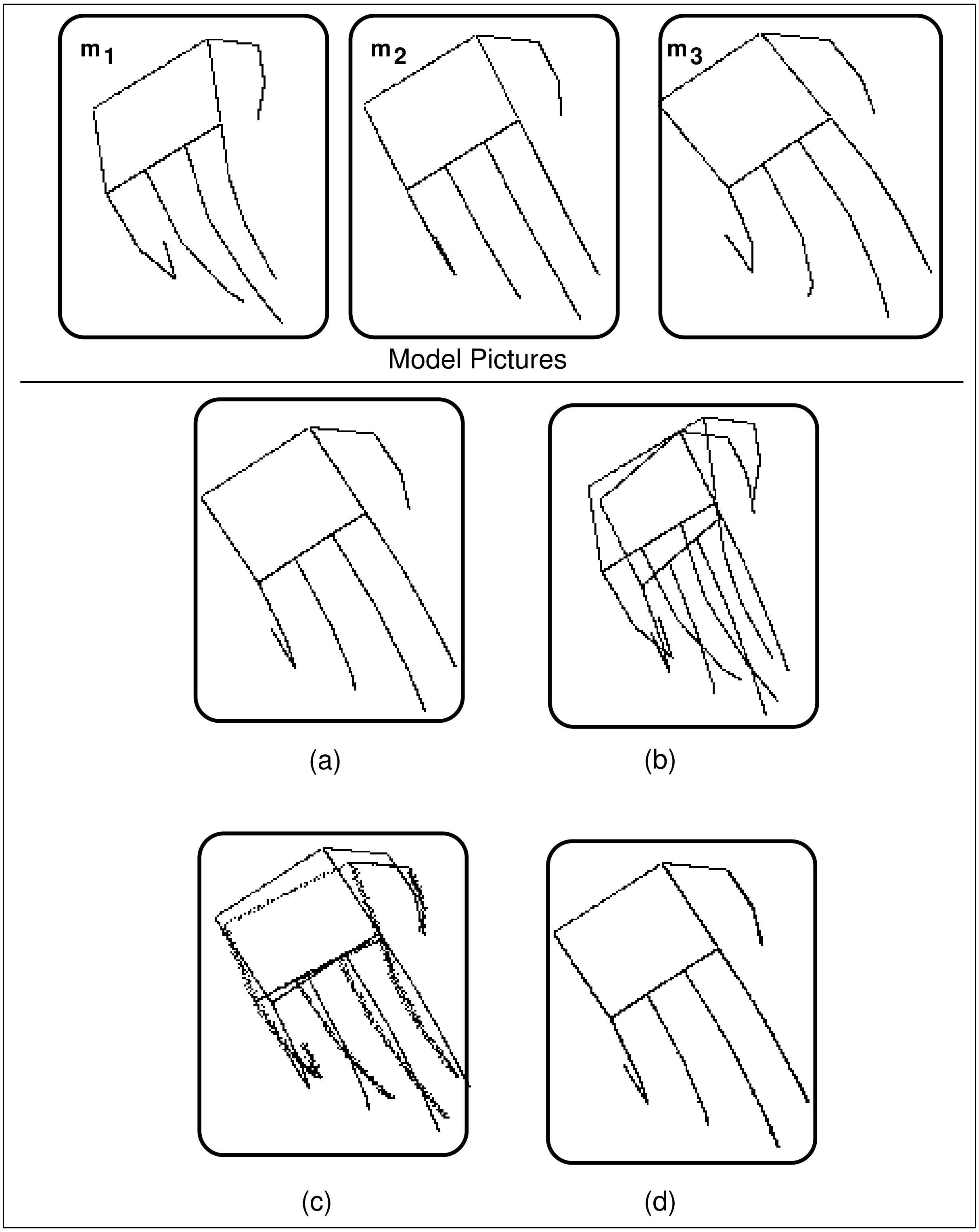

Figure 13: *Results of the correspondence technique on a three-dimensional synthetic hand. The top three pictures comprise the model. (a) The novel image* ($0.5\mathbf{m_2}+0.5\mathbf{m_3}$)*. (b) The system matched the novel image to model picture 1* ($\mathbf{m_1}$)*. The two are shown here roughly 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.*

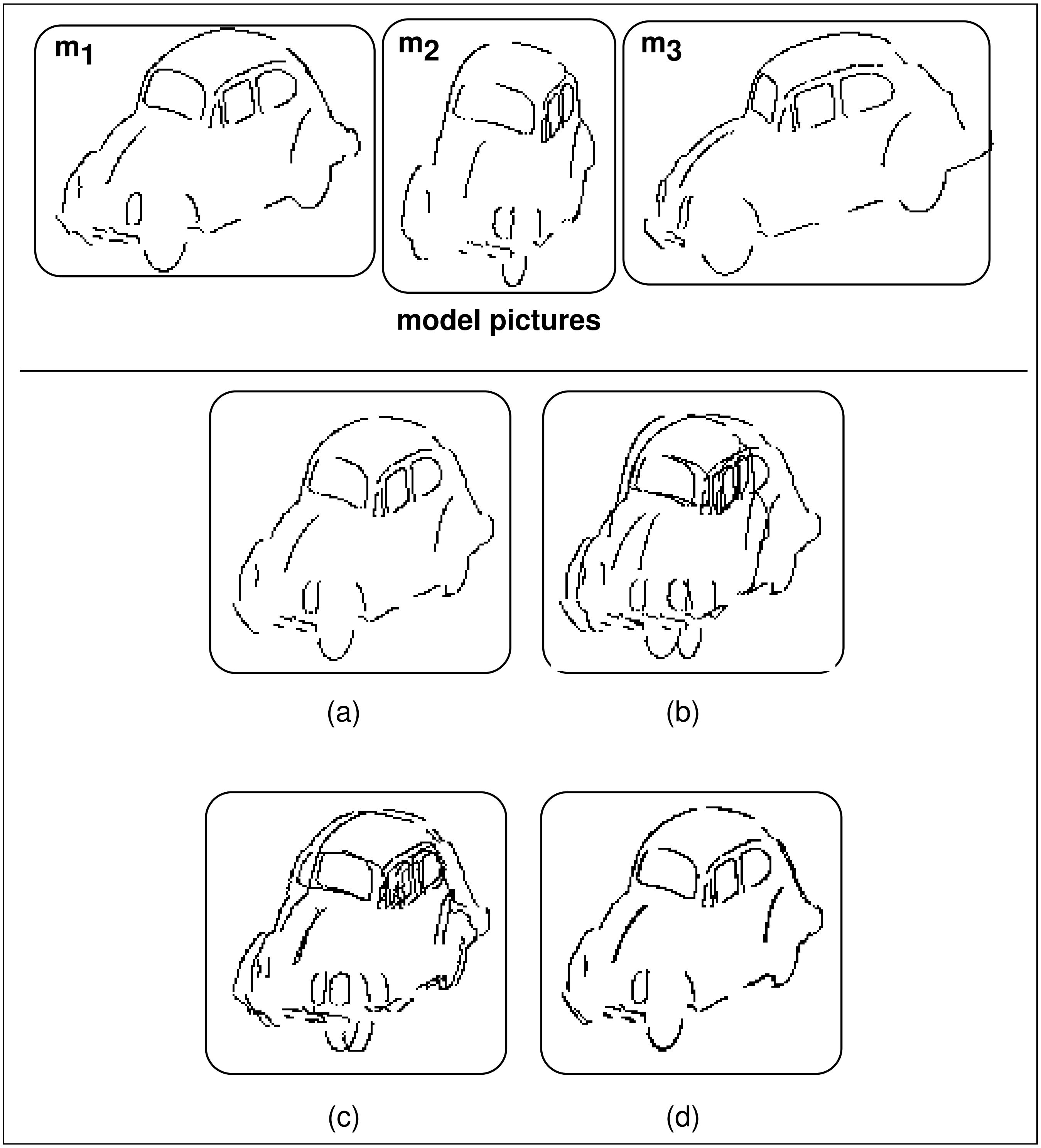

Figure 14: *Example of our recognition process on a toy VW. The model images were created by rotating the car around one axis. (a) The novel image* ($0.5\mathbf{m_1} + 0.5\mathbf{m_2}$)*. (b) We used the first model picture* $\mathbf{m_1}$ *in the matching process. A comparison of the image to this picture before the rough alignment stage 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 10 iterations.*

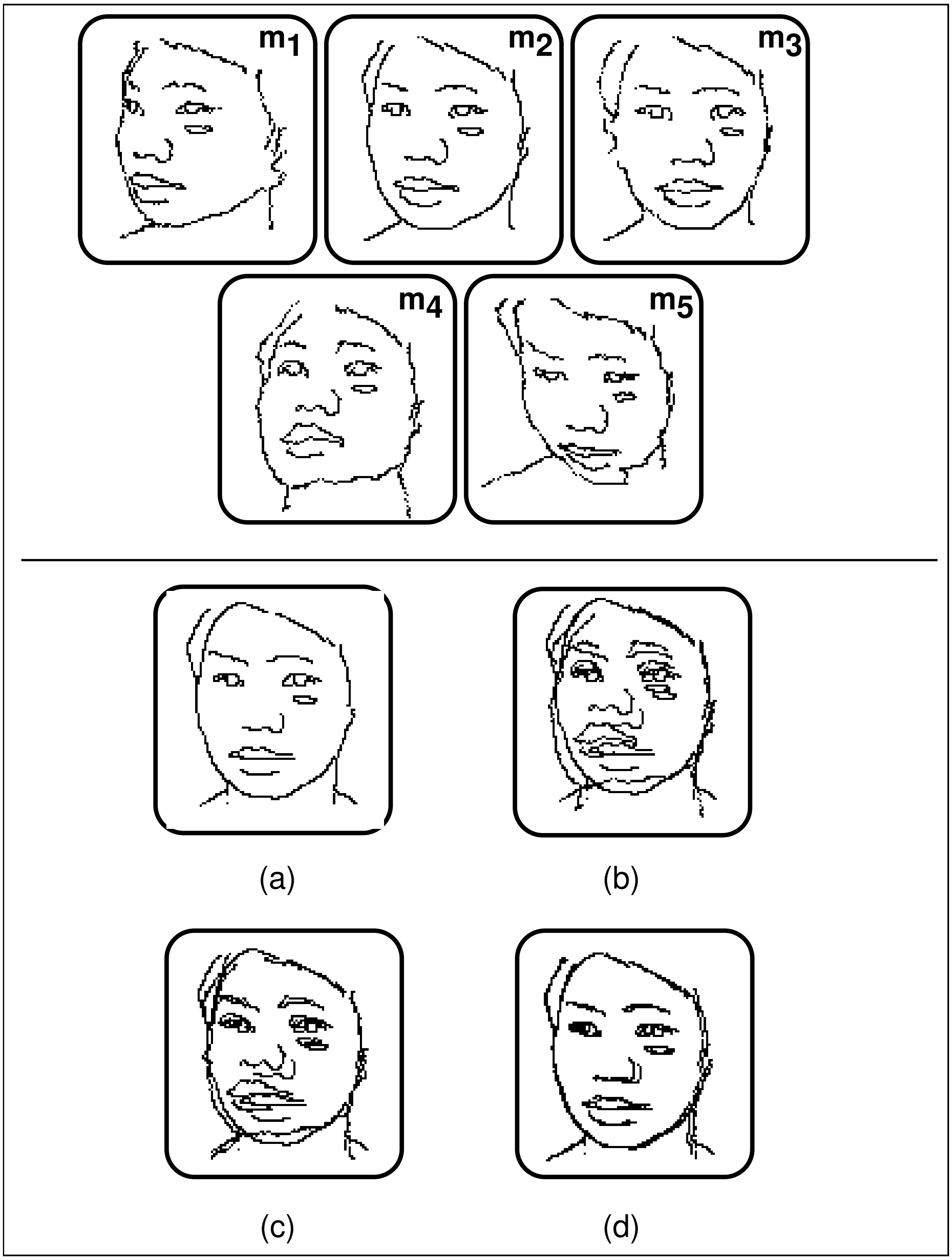

Figure 15: *Example of the correspondence process on a human face (KKS). The five model pictures of the face are shown above the horizontal line. (a) A real novel image of KKS whose face pose is between the face poses in model pictures 2 and 3. (b) The novel image is compared to model image 4. A superimposition 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.*

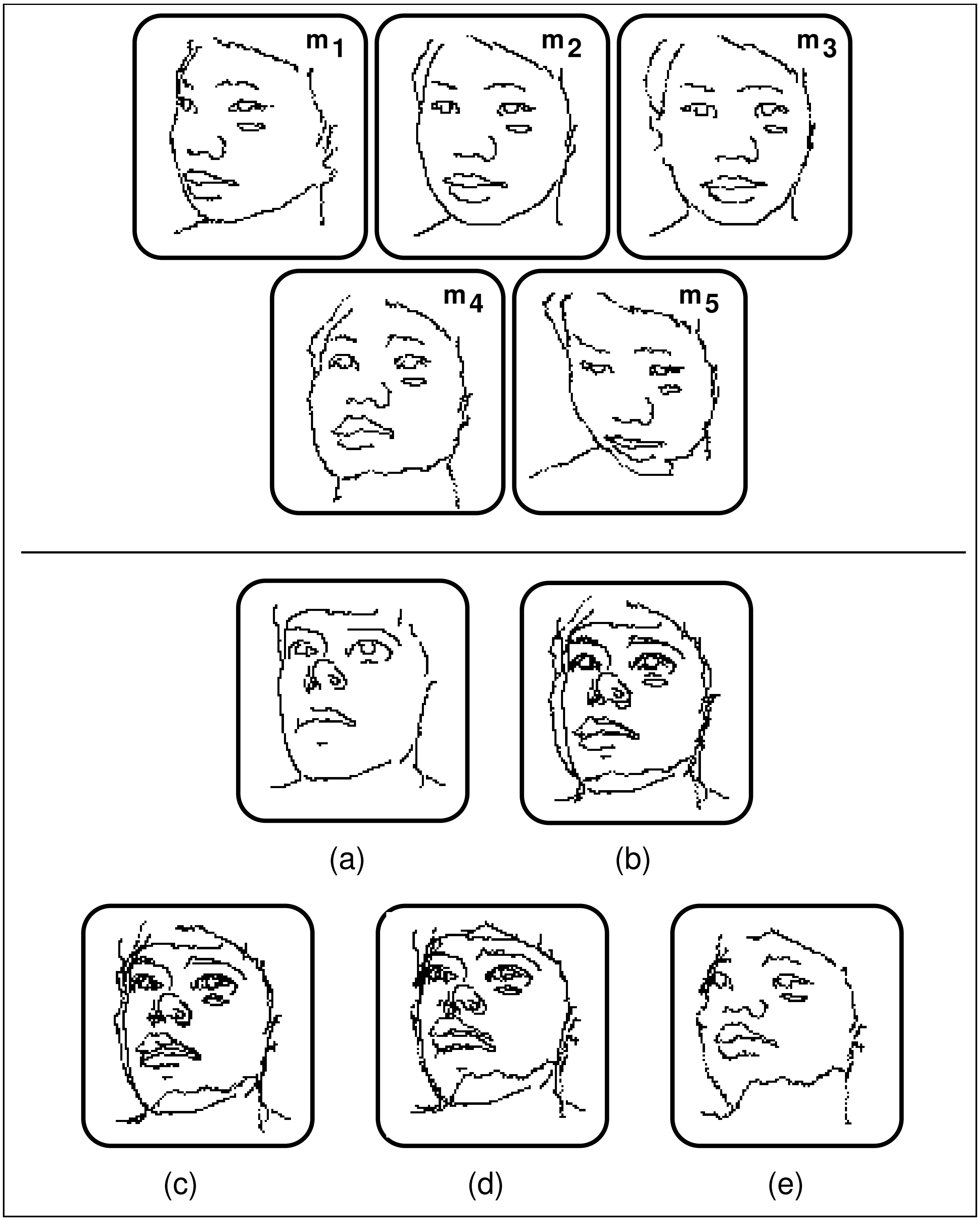

Figure 16: *Example of the correspondence process acting on a model of a face of KKS and a novel image from a different person (DB). (a) The novel image of DB. (b) A comparison of DB to KKS model image 4 prior to the matching process. (c) The result of matching the novel image to KKS model image 4 after 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*

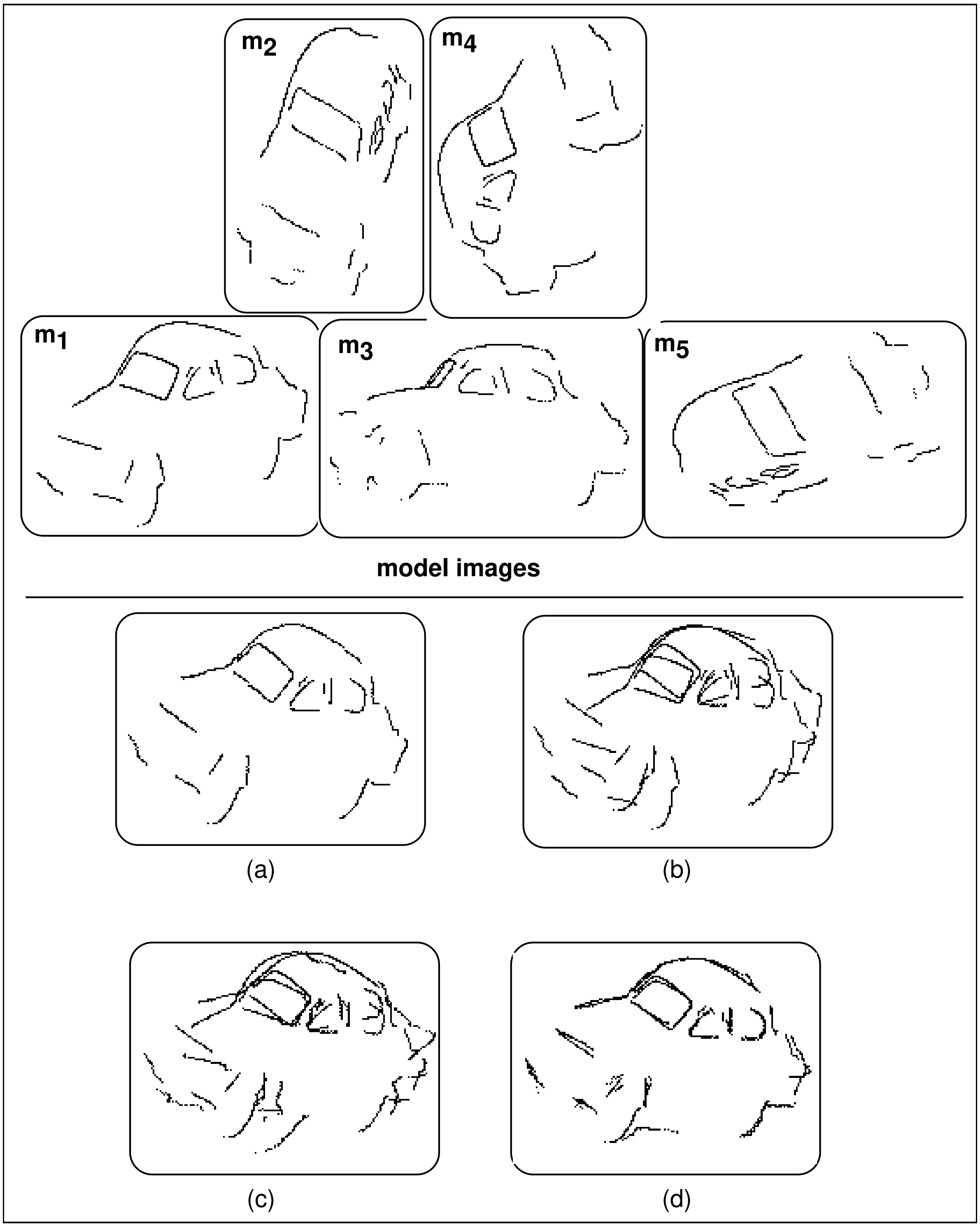

Figure 17: *Experiment using incorrect compensation for rotation in the xy plane. Five model pictures of a toy Saab are shown above the horizontal line. (a) The novel image was created by rotating model picture 1 ($\mathbf{m_1}$) by 18 degrees. (b) The correspondence process initially compares the novel image to model picture 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 10 iterations*

References

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

[2] R.A. Brooks. *Model-Based Computer Vision.* UMI Research Press, Ann Arbor, MI 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] S. Edelman and H.H. Bulthoff. Viewpoint-Specific Representations in Three Dimensional Object Recognition. A.I. Memo 1239, The Artificial Intelligence Lab, MIT, 1990.

[5] 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.

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

[7] 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.

[8] 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.

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

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

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

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

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

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

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

[16] 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.

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

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

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

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