Autonomous Robots 10, 317–338, 2001

Shape Morphing-Based Control of Robotic Visual Servoing

RAHUL SINGH
Computational Drug Discovery Group, Informatics Department, Exelixis Inc, 170 Harbor Way, South San Francisco, CA 94083

RICHARD M. VOYLES, DAVID LITTAU AND NIKOLAOS P. PAPANIKOLOPOULOS
Artificial Intelligence, Robotics, and Vision Laboratory, Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455

Abstract. We present an approach for controlling robotic interactions with objects, using synthetic images generated by morphing shapes. In particular, we attempt the problem of positioning an eye-in-hand robotic system with respect to objects in the workspace for grasping and manipulation. In our formulation, the grasp position (and consequently the approach trajectory of the manipulator), varies with each object. The proposed solution to the problem consists of two parts. First, based on a model-based object recognition framework, images of the objects taken at the desired grasp pose are stored in a database. The recognition and identification of the grasp position for an unknown input object (selected from the family of recognizable objects) occurs by morphing its contour to the templates in the database and using the virtual energy spent during the morph as a dissimilarity measure. In the second step, the images synthesized during the morph are used to guide the eye-in-hand system and execute the grasp. The proposed method requires minimal calibration of the system. Furthermore, it conjoins techniques from shape recognition, computer graphics, and vision-based robot control in a unified engineering framework. Potential applications range from recognition and positioning with respect to partially-occluded or deformable objects to planning robotic grasping based on human demonstration.

Keywords: shape recognition, shape morphing, robotic visual servoing, vision-based robot control, vision-based grasping

1. Introduction

Controlling the interactions of an eye-in-hand system with a static target is a challenging research problem. A framework towards the solution of such a problem involves identification of the desired pose at which alignment is to occur (a recognition issue), planning a trajectory for the robot to attain the desired position, planning the interactions, and careful calibration of the system and the environment. In this paper we consider the problem of grasping objects that have varying shape complexity and grasp positions by introducing a new approach to vision-based robot control based on shape morphing. In the proposed framework the recognition of the object and, thereby, the identification of its grasp position; the subsequent translational and rotational pose alignment of the manipulator with the object; and its movement in depth are controlled by using the virtual (synthetic) images from the morph. We use a model-based framework where the image of each object at a graspable pose is stored in a database. Given an unknown object in the workspace, its identity is established by morphing its contours to the shapes in the database and using a quantification of the morph as a dissimilarity measure. In the case of rigid objects (assuming correct recognition), the *virtual motion* in the morph plane (see Fig. 1) corresponds to a pose transformation with respect to a *static virtual camera*. The basic idea of the proposed method is to treat the sequence

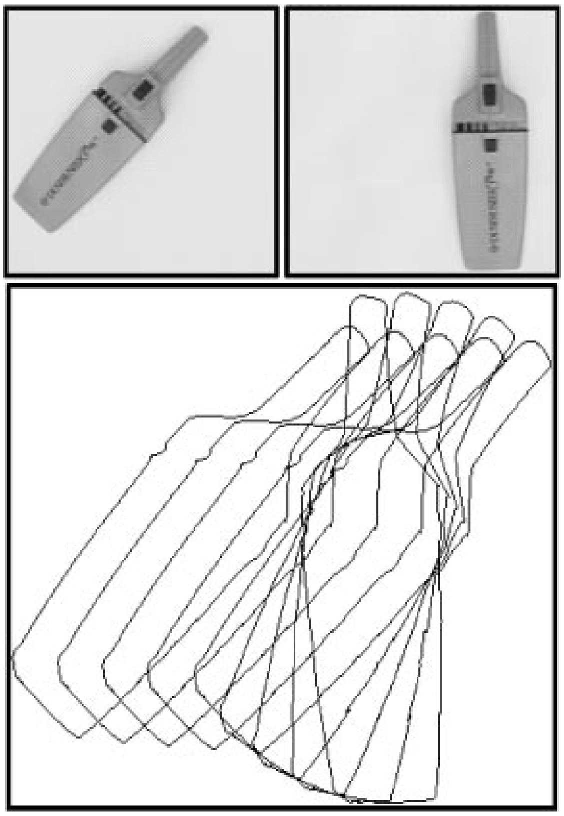


Figure 1. Starting state (top left), goal state (top right), and the virtual motion (bottom), as represented by successive synthetic images, in the morph plane.

of synthetic intermediate images as subgoals for an eye-in-hand system and guide the robot to the desired orientation and height by mapping the virtual motion in the morph plane to the actual movements of the robot in the world.

We begin this paper with an overview of the prior research in the area of vision-based grasping systems (Section 2). This is followed by the formulation of the proposed method in Section 3. Experimental results are presented in Section 4 and the work is summarized and some possible directions of future work are outlined in Section 5.

2. Issues and Prior Research

In recent literature the maturing field of robotic visual servoing is summarized among others in a tutorial (Hutchinson et al., 1996) and, in the following book (Vincze and Hager, 2000). Generally, vision-based positioning and grasping systems can be classified as *position-based* or *image-based* depending on whether the control error function is computed in the Cartesian space or in the image-plane respectively (Hutchinson et al., 1996; Vincze and Hager, 2000). The basic idea of image-based robotic visual servoing lies in defining an error between the current and the desired positions of the manipulator in image coordinates (Weiss, 1984), such that zero error implies that the desired end-effector position is reached, regardless of the camera position (Hager, 1995). In contrast, position-based methods allow the direct specification of the desired relative trajectories in the end-effector Cartesian coordinate frame.

Different methods, based on both the above have been proposed for the problem of robotic visual servoing around a static object. One of the earliest methods is a model reference adaptive control scheme that was proposed in Weiss et al. (1987). A pre-computed Jacobian was used in conjunction with an adaptive control law to control the motion of a robot-camera system to attain a certain pose with respect to a static object in Chaumette et al. (1991). In Hasimoto et al. (1992), a neural-network based approach was proposed to learn the inverse perspective transformation based on feature points. It may be noted that the above mentioned approaches were all tested in simulation or with idealized objects. A real robot (Direct Drive Arm II) was used in Papanikolopoulos and Khosla (1992), where the *controlled active vision* framework was used for positioning the robot. The *controlled active vision* framework was extended to to the full 6-DoF case with uncertain parameters in Papanikolopoulos et al. (1995). This paper also addresses the limitations and the assumptions involved in extracting 6D information from 2D images. An extension of this approach for grasping static and moving objects can be found in Smith and Papanikolopoulos (1996). Based on geometric cues obtained from the environment, a calibration free approach for manipulating a randomly placed part is formulated in Ghosh et al. (1996). In this work calibration is performed as a part of the estimation, planning and tracking process. In Hager (1995) results from projective geometry are used to design a calibration free approach. A system based on a pure position-based approach is presented in Wilson et al. (1996). A hybrid approach is proposed in Malis et al. (1998, 1999), where the control error function is computed partly in the 2D image plane and partly in the 3D Cartesian space. Allotta and Colombo (1999) discuss the use of linear models in the camera-object interaction. Their system includes a planner, a robot controller, and a visual analysis module. Gangloff, De Mathelin and Abba (1999) have presented a very fast visual servoing scheme for 3-D profile following by a 6-DoF manipulator. While their vision processing covers simple objects, their control structure is innovative compared to relevant work in the area. Janabi-Sharifi and Wilson (1998) introduce a planning strategy for vision-based grasping. Salganicoff, Ungar, and Bajcsy (1996) discuss learning strategies for vision-based grasping. Finally, visual servoing techniques have also been used in various applications (Corke et al., 1998; Joshi and Sanderson, 1998).

While much of the early research in vision-based servoing (Chaumette and Rives, 1990; Chaumette et al., 1991; Feddema and Lee, 1990; Weiss et al., 1987) involved precise off-line calibration, such systems usually suffered from lack of accuracy outside the regions where they are calibrated (Yoshimi and Allen, 1995). In contrast, recent work has focused on the use of on-line estimation of the calibration (Ghosh et al., 1996; Hager, 1995; Papanikolopoulos and Khosla, 1993; Yoshimi and Allen, 1995; Papanikolopoulos et al., 1995). Although these methods are less sensitive to calibration errors, it is worth mentioning that they primarily focus on the vision based control aspects of the problem and either do not consider formulations where the required alignment pose may vary with the object (Ghosh et al., 1996; Yoshimi and Allen, 1995) or involve manual selection of features which are used to reduce the image error and thereby control the manipulator (Hager 1995; Papanikolopoulos and Khosla, 1993; Vikramaditya and Nelson, 1999). Finally, they all formulate the error to the goal point as a standard minimization problem without trajectory control through the state space (image plane).

Robotic grasping research can be broadly categorized into two groups: empirical and analytical (Hester et al., 1999). Empirical methods mimic human selection of grasp point and pre-shape while analytical methods determine a set of contact locations that satisfy kinematic and dynamic constraints. Most of this work, both empirical (e.g., Borst et al., 1999; Voyles and Khosla, 1999) and analytical (e.g., Hester et al., 1999), has been directed toward multi-fingered hands with high complexity. Our immediate interest is in parallel-jaw grippers and pinch-type grasps.

Studies on parallel-jaw grippers have largely been analytical, due to the simplicity of considering polygons. Ferrari and Canny (1992) planned optimal grasps of hypothetical planar polygonal objects with two- and three-jaw grippers. They assumed the dimensions of the edges of the object were comparable to the width of the finger and tested all possible configurations of planar gripper and object without concern for mass properties of the object. Smith et al. (1999) included center-of-mass information to suggest a variety of ranked "grips" for the cross-section of a part under design. Polygonal cross sections on the order of tens of vertices take on the order of seconds to compute. Both these methods are based on geometric models of the polygons. Perrin et al. (2000) perform a similar analysis of grips based on real-time visual feedback in tracking real object contours using deformable snakes, but center of mass is visually estimated so asymmetric objects like hammers may fail.

An example of an empirical approach with parallel-jaw grippers is embodied in "learning by watching" (Kuniyoshi et al., 1994). They model the human hand as a parallel-jaw gripper and grasp an object in the same location that a human does. They assume the human considers possible asymmetries of the center of mass and other material properties that may affect the success of the grasp and accept the grasp point selection of the human.

3. The Proposed Method

In a realistic setting, the grasp location as well as the alignment position from which a grasp can occur, vary with each object. For example, the object in Fig. 1 is designed and balanced for grasping at the narrow handle with primary grasp forces exerted perpendicular to the major axis of the device. Attempting to grasp at the smooth, sloping centroid of the object is not robust, nor is grasping at the handle with the wrong orientation. In a generic setting, grasping any (graspable) object in the workspace, involves solving the following sub-problems:

1. Identification of the grasp points for the object.
2. Trajectory generation for the manipulator from its current position to the position from which the grasp can occur. This involves:
    - Translational and rotational positioning of the end-effector at an appropriate approach point with respect to the object.

- Movement in depth (approach) to execute the grasp.

3. Calibration between the image plane and the world.

In this paper we consider the problem of vision-directed grasping of objects based on a conceptually new framework involving image morphing. The proposed approach allows us to treat issues related to recognition, trajectory generation and on-line calibration in a unified manner. Although there are many possible ways to identify the grasp points of an object, for the purposes of this paper, we assume that the object to be grasped comes from a known set of objects (a model-based framework). Furthermore, we require a learning step where the grasp position and the desired orientation of each object, at grasp execution, are specified by the user. Thereby, each template in the model-based framework encodes object identity as well as relative object-gripper pose for grasping. This effectively equates the problem of grasp point identification with object recognition.

The aforementioned formulation induces the following operational division (similar to Malis et al., 1998), of the the positioning and grasping problem:

1. *Off-line learning*: The eye-in-hand system is moved to the alignment position with respect to an object. The alignment is such that the gripper is directly above the grasp position of the object. The view of the object is captured and stored in an image database.
2. *Visual servoing*: After the camera and/or the target has moved, the camera motion is controlled so that the visual error between the current view of the object and its reference view (obtained in the previous step) is reduced to zero.

The first step creates a database of images where each image represents the desired view of the object and specifies the pose which the manipulator must attain to approach and grasp the object. The alignment and grasp positions can be arbitrarily defined and are neither limited by assumptions on object shape, nor depend on hand-crafted rules for their definition.

During the second step, a given unknown input object is compared at run time with the models acquired and stored in the database by using shape morphing. The comparison is based on determining a quantification of the non-rigid deformations of the input object contours to the models stored as templates. Of primary importance, in the present context, is the sequence of *virtual images* generated during the morph that describe the progressive transformation of the input (both in terms of its shape and pose) to the template (see Fig. 1). These images are used as subgoals to guide an eye-in-hand robotic system to attain the desired pose and then grasp the identified object in its workspace.

In summary, the proposed framework treats the problem of vision-based robotic grasping in terms of solutions to the following sub-problems:

1. *The recognition problem*: Determination of the object's identity and (by definition) its alignment position and the object-relative grasp location.
2. *Translational and rotational alignment*: Positioning the end-effector with respect to the target in terms of translation and rotation.
3. *Grasp execution*: Controlling the movement of the end-effector in depth so as to execute the grasp.

3.1. The Recognition Problem

Let $\mathbf{S}^I = [S_0^I, \ldots, S_n^I]$ and $\mathbf{S}^T = [S_0^T, \ldots, S_n^T]$ be the point sets representing the input and the target contours, respectively. The morph of shape $\mathbf{S}^I$ to shape $\mathbf{S}^T$ is defined by a sequence of intermediate shapes, that are generated by a cross dissolve operation on the corresponding points of the two shapes:

$$\begin{aligned}\mathbf{S}(t) &= u\mathbf{S}^I + t\mathbf{S}^T \\ &= \left[uS_0^I + tS_0^T, uS_1^I + tS_1^T, \ldots, uS_n^I + tS_n^T\right] \\ &= [S_0(t), S_1(t), \ldots, S_n(t)] \qquad (1)\end{aligned}$$

where $u = 1 - t$. $S_i(t)$ is the ith contour point in the intermediate shape, formed at time t. The time parameter t is normalized to the interval [0, 1]. In practice, the contours of the object are approximated by using a segmentation algorithm. In this work, we use the following segmentation technique: Given a contour, the algorithm (Ray and Ray, 1993) is applied to get its piecewise linear approximation. For each segmentation point thus obtained, the curvature is computed and points with high curvature are retained. A merging step is then executed to remove points due to noise or quantization errors. The proposed segmentation technique provides low approximation errors and consistent placement of segmentation points. Details of this technique and its performance can be found in Singh and Papanikolopoulos (2000). It may be pointed out that if the input objects undergo substantial deformations (examples include hand-drawings or objects with significant occlusions), it may be desirable to use a segmentation algorithm that is not based on reducing an error norm, as we have chosen. Segmentation strategies based on the automatic detection of perceptually important points could be used in such cases (Brault and Plamondon, 1993; Pavlidis et al., 1998).

Since the contours $\mathbf{S}^I$ and $\mathbf{S}^T$, in general, have different number of segmentation points, for the morph (as described by Eq. (1)) to occur, a point correspondence between the input and the target shapes is required, such that every segmentation point on the input contour corresponds to at least one segmentation point on the target contour and vice versa. To find an optimal correspondence (and thereby an optimal morph), we follow a strategy that is motivated by the physics-based approach to shape blending (Sederberg and Greenwood, 1992) and define the cost of a point correspondence as the sum of stretching and bending energies required to bring about the correspondence. The stretching energy is computed for every segment (pair of points) and is defined as:

$$E_s = k_s \frac{(L_T - L_O)^2 - (L_I - L_O)^2}{(1 - c_s)L_{\min} + c_s L_{\max}} \tag{2}$$

where

$$L_{\min} = \min(L_O, \ldots, L_I, L_T)$$

and

$$L_{\max} = \max(L_O, \ldots, L_I, L_T)$$

In Eq. (2) E_s denotes the stretching energy spent in the current deformation, L_O, L_I, and L_T denote the segment lengths at the beginning, before the current deformation, and after the current deformation, respectively. The term c_s corresponds to the penalty for segments collapsing to points and k_s is the stretching stiffness parameter. The bending energy E_b is computed for point triplets and denotes the cost of angular deformation.

$$E_b = |k_b[(\phi_T - \phi_O)^2 - (\phi_I - \phi_O)^2]| \tag{3}$$

where k_b indicates bending stiffness, ϕ_O represents the original angle, and ϕ_I and ϕ_T denote the angle before the current deformation and the angle after the current deformation, respectively.

The optimal morph, between two contours is defined by the correspondence requiring the least stretching and bending energy. By constraining the deformations at the segmentation points, the following optimal substructure property may be observed: The optimum cost of the point correspondence (S_i^I, S_j^T) equals the optimum cost of the previous point correspondence (S_{i-1}^I, S_j^T) or (S_{i-1}^I, S_{j-1}^T) or (S_i^I, S_{j-1}^T) and the cost of establishing the correspondence (S_i^I, S_j^T). Based on the above, an efficient $(O(mn))$ dynamic programming scheme can be constructed for morphing a contour C_A with m points to another C_B having n points. Since the energy computation described above, requires a starting point correspondence, we define the optimal morphing between two contours C_A and C_B as:

$$D_{\text{morph}}(A, B) = \min_{\Omega} E(C_A, C_B) \tag{4}$$

Here, Ω denotes the set of all starting point correspondences between the contours C_A and C_B.

In Fig. 2, an example of the adjustment in the number of segmentation points is shown for the contour of an object. The optimal starting correspondence obtained from the morph is depicted by the unbroken arrow. Examples of segmentation point adjustment are shown with dashed arrows. The reader may note, that the correspondences associated with the segmentation point adjustments are optimal in the sense that they minimize the stretching and bending energies

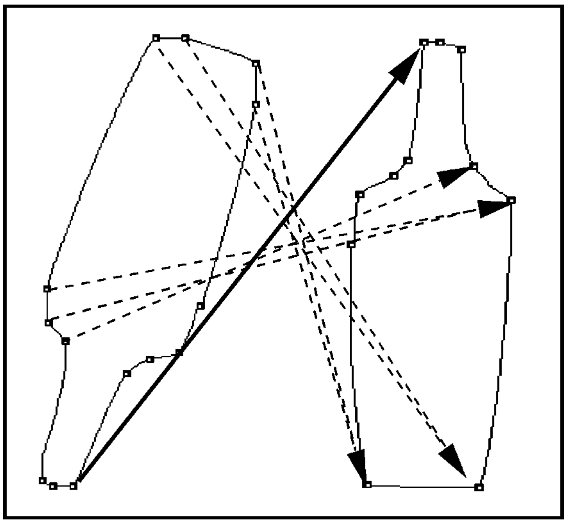


Figure 2. Adjustment of the number of segmentation points during the morph.

required to transform the input shape to the target shape.

Owing to the fact that the energy measures (Eqs. (2) and (3)) used in the morph depend on the relative position of the contour points of an object and not their absolute position, they are invariant to translation and rotation. Invariance to scale changes is achieved by mapping the shapes to a unit square. This allows for correct recognition in cases when the training sample (template) is captured at a different height than that at which recognition and grasping experiments are conducted.

The formulation of the morph, as defined above, uses a linear cross dissolve operation (Eq. (1)). Therefore, it does not, by itself, guarantee *physically valid* intermediate shapes between images of the same object, unless the input and the target contours have rotational alignment. Specifically, the lack of physical validity may be manifested by a cross-over of the object contours during the intermediate morphs. (Imagine the segmentation points in Fig. 2 switching places gradually from initial image to final image. The respective trajectories (denoted by the dotted arrows) would cross over at the mid point, creating invalid intermediate images.) This may be avoided by including in the morph a rectification of the rigid transformations (translation and rotation) between the input and the target shapes, by warping them prior to the application of Eq. (1). The rectification of the rigid transformations is based on the observation that, during the recognition process, the computation of the optimal morph provides a point correspondence which is invariant to translation and rotation of the corresponding objects. This correspondence can therefore be used to estimate the elongation vectors for the input and the target shapes. The translational discrepancy between the centers of the input and the target elongation vectors is used for translational rectification between the images. The rotation between the two shapes is estimated by computing the angle between the elongation vectors, about their center, after translation rectification. Correction of the rotational discrepancy occurs by generating virtual images in the direction of the smaller angle of rotation. The rectification of the rigid transformations (translation and rotation) involves an update of the coordinates of the input and does not affect the correspondences that were obtained during the computation of the minimal energy. These correspondences along with the updated coordinate values of the input are used to rectify the deformations between it and the target.

3.2. Translational and Rotational Alignment using Synthetic Images

Let $\{W\}$ denote the world frame, $\{R\}$ the robot end-effector frame, and $\{C\}$ the camera tool frame. Let also the morph plane be denoted by $\{I\}$ and the object frame by $\{O\}$. The relations between these coordinate frames can be described by the following homogeneous transformations:

- ${}^{W}_{R}T_i$: Describes the pose of the end-effector in the world frame at time instant i. This transformation is known at any time instant.
- ${}^{R}_{C}T$: The constant transformation between the robot end-effector and the camera (unknown).
- ${}^{C}_{O}T_i$: Denotes the object in the camera frame and is unknown.
- ${}^{C}_{I}T$: The image projection transformation. This transformation is assumed known from an off-line calibration step, but it could be automatically deduced. In the equations below, we use ${}^{I}_{C}T$ which is the inverse of the transformation ${}^{C}_{I}T$.
- ${}^{I}_{O}T_i$: The transformation describing the object in the image plane at time instant i. This transformation is assumed to be known.
- ${}^{W}_{O}T$: The transformation between the object frame and the world frame which is unknown.

A graphical depiction of these transforms and their interrelationship is shown in Fig. 3. The series of transformations may be arranged in two loops, denoted as A and B, respectively. These loops are coupled mathematically by the transform ${}^{C}_{O}T_i$. The subscript i is omitted from the transformations for brevity, when from the context it is clear that the relation holds across all time instants.

The object frame can be related to the world frame and the image frame based on the following relations:

$$ {}^{W}_{O}T = {}^{W}_{R}T \; {}^{R}_{C}T \; {}^{C}_{O}T \tag{5} $$

$$ {}^{I}_{O}T = {}^{I}_{C}T \; {}^{C}_{O}T \tag{6} $$

Denoting the initial and the goal transformation between the object and the image frames by ${}^{I}_{O}T_0$ and ${}^{I}_{O}T_{\text{goal}}$ respectively, we have from loop B of Fig. 3

$$ {}^{I}_{O}T_0 = {}^{I}_{C}T \; {}^{C}_{O}T_0 \tag{7} $$

Let the transformation from the initial state to the goal state in the morph plane be denoted as ${}^{I}_{M}T_{\text{goal}}$. Letting

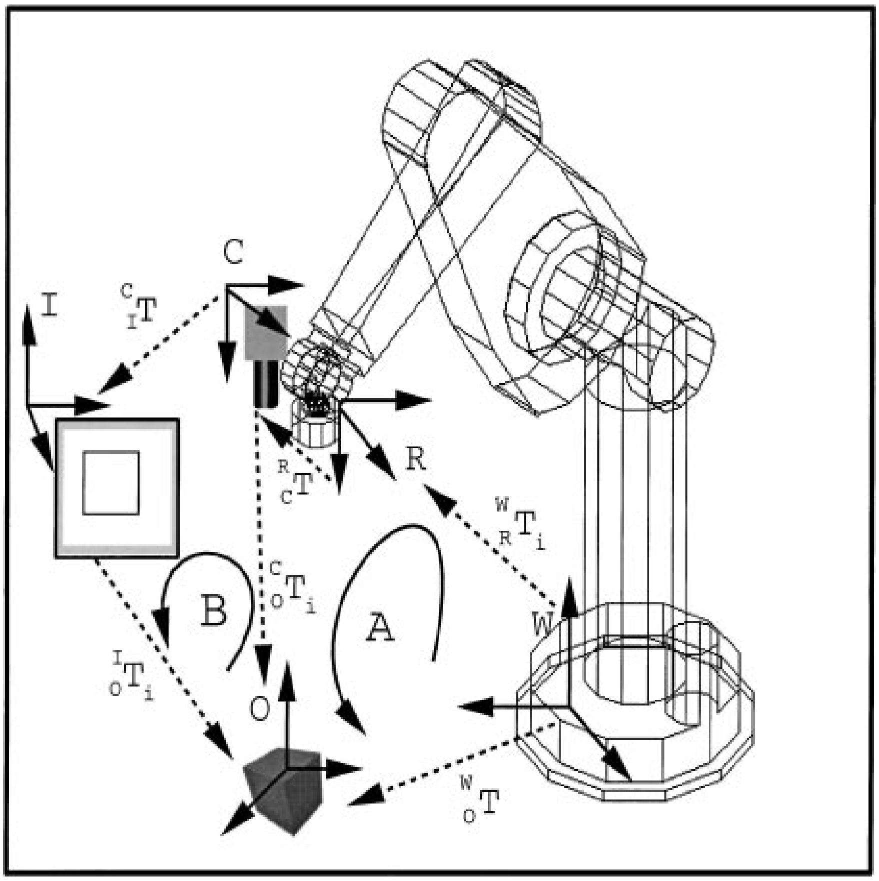


Figure 3. Various coordinate frames and their relationships.

the intermediate transformation between the camera and the object to be ${}^C_O T_i$, we obtain the following relationships

$$ {}^I_O T_{\text{goal}} = {}^I_M T^{-1}_{\text{goal}}\, {}^I_O T_0 = {}^I_C T\; {}^C_O T_i \tag{8}$$

Replacing from Eq. (6), the value of ${}^I_O T_0$ in Eq. (8) and solving for ${}^C_O T_i$, we get

$$ {}^C_O T_i = {}^I_C T^{-1}\; {}^I_M T^{-1}_{\text{goal}}\; {}^I_C T\; {}^C_O T_0 \tag{9}$$

But from Eq. (5), we have for ${}^C_O T_0$

$$ {}^C_O T_0 = {}^R_C T^{-1}\; {}^W_R T_0^{-1}\; {}^W_O T \tag{10}$$

Replacing this value in the previous equation, we get

$$ {}^C_O T_i = {}^I_C T^{-1}\; {}^I_M T^{-1}_{\text{goal}}\; {}^I_C T\; {}^R_C T^{-1}\; {}^W_R T_0^{-1}\; {}^W_O T \tag{11}$$

Let us now consider the relationship described in loop A of Fig. 3 and expressed by Eq. (10). Let ${}^R T_{i_d}$ be the desired motion of the robot at the intermediate stage i. We then have

$$ {}^W_R T_i = {}^W_R T_0\;\; {}^R T_{i_d} \tag{12}$$

Replacing in Eq. (10), the value of ${}^W_R T_i^{-1}$, from Eq. (12), we obtain the following relationship for ${}^C_O T_i$

$$ {}^C_O T_i = {}^R_C T^{-1}\;\; {}^R T^{-1}_{i_d}\;\; {}^W_R T_0^{-1}\; {}^W_O T \tag{13}$$

From the above equation, we obtain the following expression for ${}^R T^{-1}_{i_d}$:

$$ {}^R T^{-1}_{i_d} = {}^R_C T\;\; {}^C_O T_i\;\; {}^W_O T^{-1}\;\; {}^W_R T_0 \tag{14}$$

Using the value of ${}^C_O T_i$ from Eq. (11), replacing it in the above equation and simplifying, we finally have

$$ {}^R T_{i_d} = {}^R_C T\;\; {}^I_C T^{-1}\;\; {}^I_M T_{\text{goal}}\;\; {}^I_C T\;\; {}^R_C T^{-1} \tag{15}$$

If the calibration transform ${}^{R}_{C}T$ was known, Eq. (15) could be used to align the eye-in-hand system at the desired position with the object. However, we have assumed no knowledge of this calibration matrix. Instead, we estimate it on-line using the intermediate images that result from the morph sequence. We start with an arbitrary estimation of ${}^{R}_{C}T$ and use each intermediate virtual image as a sub-goal. The deviation between the actual and the desired trajectory is then computed by morphing the actual view of the object to the current virtual image being used as a sub-goal. The error transformation thus obtained is used to update (with possible averaging), the estimation of ${}^{R}_{C}T$. The translational displacement, it may be noted, can only be estimated up to a scale factor. We therefore, follow an approach similar to that in Malis et al. (1998) and estimate the scale factor using the ratio of the desired move to the actual move. In the current implementation we first align the end-effector in terms of the translation and then account for the rotations. The latter step may be followed by a set of translational corrections if the grasp position of the object is not around its centroid (in such a case executing the rotations may induce a translational misalignment). It is possible, to interleave the positional alignment steps with recognition, where the real view of the object after each move is morphed to the database to verify the object identity. This setup is useful in situations where the workplace may have multiple, possibly self-occluding, objects or in cases where the distance of the object from the end-effector causes severe contour distortions and may lead to incorrect initial recognition. Another important parameter in the proposed framework is the number of intermediate synthetic images generated. Using very few intermediate images would lead to non-smooth manipulator trajectories and slow convergence of the calibration estimation, while using too many images would slow down the manipulator motion. In our experiments, typically, the morph between any two shapes (real-real or real-synthetic) was uniformly sampled at five instances, to give five intermediate synthetic images.

3.3. *Grasping using Virtual Images*

After the translational and rotational alignment has occurred, the end-effector is positioned such that the desired trajectory for the grasp can be achieved by moving the manipulator along the **Z**-axis of the end-effector frame. However, control of the entire downward motion till the grasp point by using contour morphing is infeasible due to the fact that at close proximity of the end-effector to the object, the object contour falls outside the image window. A possible solution to this is the use of artificially placed features inside the object regions (Malis et al., 1998; Smith and Papanikolopoulos, 1996). Such an approach may however, constrain the applicability of the system. In our work we solve this problem by storing the distance-to-grasp along with the image of the object during the off-line learning phase. Typically, the distance-to-grasp is measured and stored from the minimal height of the gripper over the object. The minimal height is determined as the closest vertical proximity of the gripper to the object at which the object's image is completely contained in the processing window. It may be of interest to note that this type of information can also be derived from systems that are built around the *programming by human demonstration* paradigm, where an instrumented human demonstrates the task to be performed and via points are automatically extracted (Voyles and Khosla, 1999). The framework used by us to control the motion of the manipulator along the **Z**-axis of the end-effector frame is not limited to the distance-to-grasp information collected from a minimal height above the object. Specifically, the test case may have a manipulator at a position that is closer to the object than the training case where the distance-to-grasp information was collected (as long as the object contour is in the image window). An example of this situation is presented in the experimental results (Section 4).

Our approach towards controlling the movement of the robot along the **Z** axis is based on the concept of the *focus of expansion* of the camera (Horn, 1986). This is justified by the fact that the robot motion after pose-alignment is along the **Z** axis of the end-effector frame, which coincides with the optical axis of the camera. The focus of expansion, thus is the center of the image plane. In this setting, the rate of expansion of the image is proportional to the distance of the end-effector from the object. Controlling the movement in the **Z** axis is based on the above observation. A sequence of images is synthesized by morphing the current view of the object and the view stored in the database without size normalization. The primary change exhibited in the morph is in the size of the synthesized images, where the size of a real or virtual image is computed as the number of pixels that fall inside its contour. The motion of the manipulator is guided by comparing the size of the object in the image plane with the size of the corresponding synthesized image. The transformation that maps the change in the image size to the motion of the robot is estimated on-line by using the relationships derived in the following:

Let $\{I\}$ and $\{R\}$ denote the morph plane and the robot end-effector frame respectively. Further, let ${}^{R}_{I}T^{\mathbf{z}}$ denote the true (unknown) transform from $\{I\}$ to $\{R\}$, that maps the rate of expansion of the image to the movement of the robot in **Z**. Let ${}^{R}_{I}\tilde{T}^{\mathbf{z}}$ be its current estimate of this mapping. If ${}^{I}E_d$ is the desired size change in the image plane, we have

$$ {}^{R}\tilde{T}^{\mathbf{z}}_{d} = {}^{R}_{I}\tilde{T}^{\mathbf{z}}\ {}^{I}E_d \tag{16}$$

where ${}^{R}\tilde{T}^{\mathbf{z}}_{d}$ is the actual motion command sent to the robot based on the current estimation of the mapping ${}^{R}_{I}\tilde{T}^{\mathbf{z}}$. The resultant change ${}^{I}T_a$, in the image size due to this motion can be defined as:

$$ {}^{I}T_a = {}^{R}_{I}T^{\mathbf{z}^{-1}}\ {}^{R}\tilde{T}^{\mathbf{z}}_{d} \tag{17}$$

Let ${}^{I}T^{\mathbf{z}}_{\text{error}}$ be the error defined in the image plane. ${}^{I}T^{\mathbf{z}}_{\text{error}}$ is computed after every move of the robot in **Z**. We define this error as the absolute value of the difference between the expected and the actual size of the object in the image plane. ${}^{I}T^{\mathbf{z}}_{\text{error}}$ may therefore be expressed as follows:

$$ {}^{I}T^{\mathbf{z}}_{\text{error}} = {}^{I}E_d - {}^{R}_{I}T^{\mathbf{z}^{-1}}\ {}^{R}\tilde{T}^{\mathbf{z}}_{d} \tag{18}$$

Note that the expected image size may be defined either by using the intermediate virtual images generated during the morph, or by using the template image. Combining Eqs. (16) and (17) we get,

$$ {}^{I}T^{\mathbf{z}}_{\text{error}} = {}^{R}_{I}\tilde{T}^{\mathbf{z}^{-1}}\ {}^{R}\tilde{T}^{\mathbf{z}}_{d} - {}^{R}_{I}T^{\mathbf{z}^{-1}}\ {}^{R}\tilde{T}^{\mathbf{z}}_{d} \tag{19}$$

By rearranging terms we get the error between the inverses of the estimated and actual mappings:

$$ {}^{I}T^{\mathbf{z}}_{\text{error}}\ {}^{R}\tilde{T}^{\mathbf{z}^{-1}}_{d} = {}^{R}_{I}\tilde{T}^{\mathbf{z}^{-1}} - {}^{R}_{I}T^{\mathbf{z}^{-1}} \tag{20}$$

So the new estimate of the mapping, ${}^{R}_{I}\tilde{T}^{\mathbf{z}'}$, becomes

$$ {}^{R}_{I}\tilde{T}^{\mathbf{z}'} = \left({}^{R}_{I}\tilde{T}^{\mathbf{z}^{-1}} - {}^{I}T^{\mathbf{z}}_{\text{error}}\ {}^{R}\tilde{T}^{\mathbf{z}^{-1}}_{d}\right)^{-1} \tag{21}$$

which we can use either directly to replace the old estimate or perform some filtering to update the estimate in a more controlled fashion to reduce sensitivity to noise.

In Fig. 4 the manipulator views during different stages of the positioning in depth for a tape dispenser are shown. The desired view of the object corresponding to the height the manipulator needs to attain, is shown in black borders in the lower row. The view at the starting state (after translational and rotational alignment), is shown in the leftmost image of the top row. In the proposed method, the mapping between the required change in the image size and the corresponding movement in depth, is learnt on-line. This may cause the manipulator movement in depth to be non-monotonic. An example is the given case, where the first and the third moves along the **Z** axis, bring the end-effector closer to the object than is desired. The movement of the gripper for this case is plotted in Fig. 13.

In summary, the proposed approach of controlling manipulator interactions with objects using shape morphing, consists of the following steps:

1. *Shape recognition*: The shape model, most similar to the input, in terms of the stretching and bending energies is determined. The identity of the closest shape is obtained by using the optimal substructure property described above.
2. *Rectification of rigid transformations*: The elongation vectors for the input and the target shapes are computed. These are used to recover and rectify the translational and rotational differences. The coordinates of the input shape are updated to reflect the rectification.
3. *Rectification of deformations*: The updated coordinate values of the input along with the correspondences computed during the recognition stage are used for rectifying the deformations using Eq. (1).
4. *Translational and rotational alignment*: Based on the virtual images generated during the morph the gripper is aligned over the grasp position of the object. The motion of the end-effector is controlled using the relation described in Eq. (15).
5. *Grasp execution*: After the translational and rotational alignment is complete, a morph sequence is generated without size normalization. Based on the changes in the size of the virtual images, the motion of the manipulator is controlled using Eq. (21). On reaching the minimal attainable height over the object, the distance to grasp information stored in the database is used to execute the final approach.

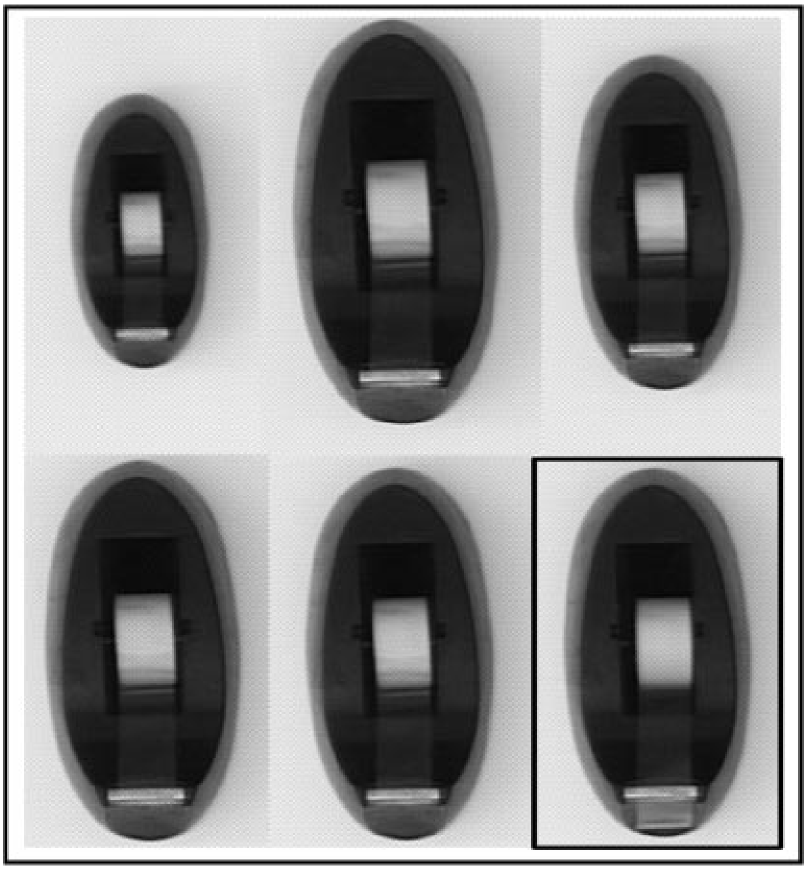


Figure 4. Different stages during the positioning of the manipulator in depth for grasping a tape dispenser.

4. Experimental Results

The method has been used for positioning a PUMA 560 manipulator with respect to planar objects as well as positioning and grasping 3D objects using their planar projections. The experimental setup consisted of an IRIX Indigo workstation for generating the morphs. The results from morphing were communicated to the eye-in-hand system, which consisted of a PUMA 560 manipulator with a custom real-time controller to circumvent VAL.

A set of sixteen objects having varying shape complexity were used in the recognition and grasping experiments. These objects are shown in Fig. 6. Positioning and grasping experiments were conducted on a subset of six objects chosen from this set, which are shown in Fig. 5. These objects were specifically selected to test the system's ability to servo objects

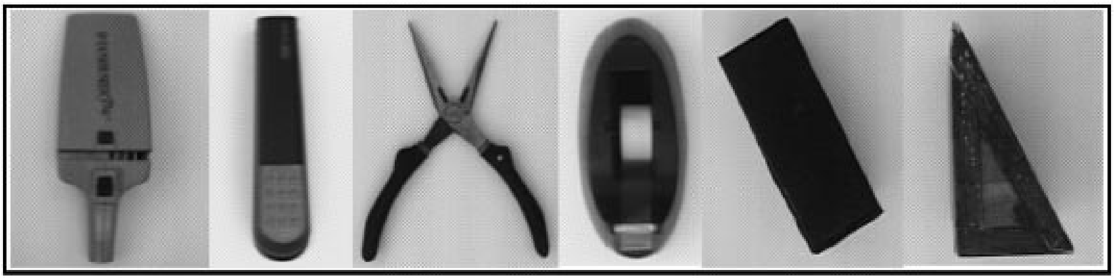


Figure 5. Objects used in the pose alignment and grasping experiments.

with similar contours (object 2 and object 5), objects having non-convex shapes (object 1 and object 3), objects whose shapes may not be perfectly rigid (object 6, where the pliers may be open to varying extents), and objects that can only be grasped at precise locations due to shape, size, or functionality constraints (object 1 at the top of the handle, object 2 at the top of the stapler, and object 4 precisely at the middle, around the tape).

During the off-line learning phase, a single image of each object at a graspable pose was stored as a template in the image database and the image was annotated with the corresponding distance-to-grasp information. The positioning and grasping experiments were conducted by placing the eye-in-hand system over the workspace, looking down. For the grasping experiments, the starting height of the manipulator was varied for each object. The objects were randomly placed at a different position and orientation within the field of view of the eye-in-hand system. For objects like the pair of pliers, non-rigid shape changes like opening them to different extents were also included. The contour of each object was extracted after moment-preserving thresholding and segmented using the algorithm outlined in Section III-A. The value of its morph to the templates in the database, was used as a dissimilarity measure to determined its identity. We note that in the current setting only one object was presented to the robot at a time and therefore the decision regarding which object to servo was not required. In a setting involving multiple objects, the current framework can be applied by extracting each blob corresponding to a given object, followed by contour detection and then morphing the contour-based shape descriptions to the templates in the database.

In Table 1, the average dissimilarity measure values over test instances for each object from our database are shown. The values in the table represent the average quantification of the optimal energy required for transforming each *test shape* to the corresponding stored *template*. The lowest value of the dissimilarity measure (energy spent in morphing), indicating the best match is underlined. For the given test instances, no mis-recognitions occurred. For the complete set of objects depicted in Fig. 6, an overall recognition rate of over 98 percent was obtained by using this method. Analysis and details of the recognition experiments on this set of objects as well as other deformable shapes may be found in Singh and Papanikolopoulos (2000). There, the reader can also find a formal analysis of the metric properties of the morph-based dissimilarity measure. In Singh and Papanikolopoulos (2000), we describe how the metric property, used in conjunction with predefined reference shapes can significantly reduce the recognition time by allowing retrieval of the correct template without an exhaustive search of the database. The basic idea lies in relating templates in the database to a predefined reference shape. The morph of a test shape to the reference shape then provides a criterion (based on the triangle inequality) for eliminating templates that are vastly different from the input without actually computing the morph between them. The processing and recognition time, using the above idea, was less than one second per test shape, for the entire set of templates (shown in Fig. 6), with the current hardware.

After the identity of an input object was established from the morph, the images generated during the morph were successively used as sub-goals to guide the manipulator to the correct alignment position. After each move of the manipulator, the image of the object obtained from the current position was morphed to the virtual image serving as the current sub-goal. The corresponding morph value (a *sub-morph*), in this case, indicated the local error in the image plane of the trajectory sub-goal. The pose difference between the two was used to update the estimated calibration matrix and

Table 1. Recognition results for the set of objects.

Test shapes	Templates					
	1	2	3	4	5	6
1	257.87	672.65	862.61	648.74	585.11	686.62
2	512.91	64.45	1296.70	125.87	148.10	215.85
3	997.98	1269.90	249.26	1079.30	1137.70	1178.50
4	579.06	207.46	1044.50	90.73	205.02	253.35
5	454.38	119.63	1169.20	155.61	39.08	101.14
6	637.74	193.19	1282.70	199.74	129.43	64.38

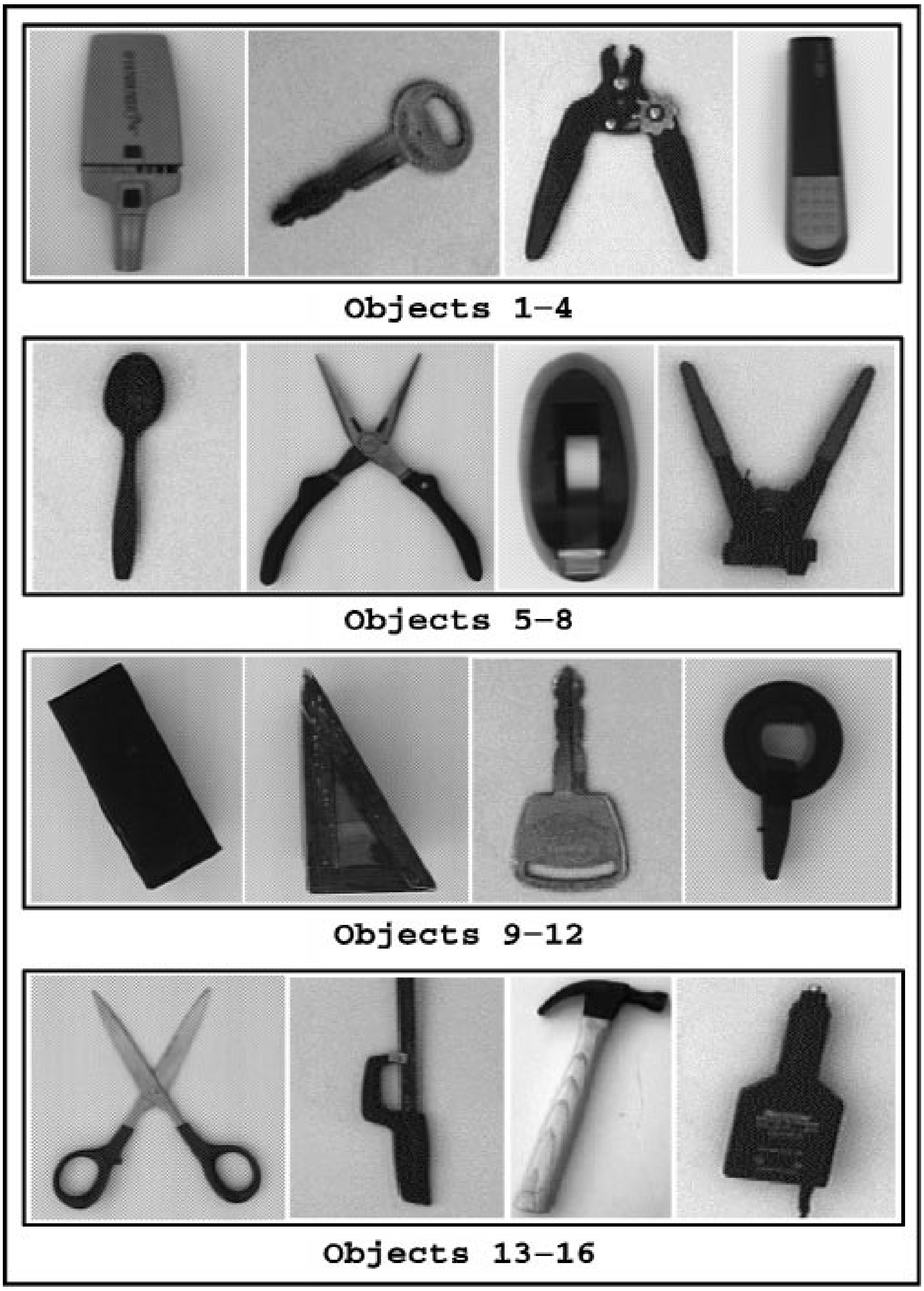


Figure 6. The set of all objects used in recognition experiments (of which six are used in the positioning and grasping experiments described in this paper).

move the manipulator to a new position. When the pose difference between the actual image and the virtual images became less than a threshold (empirically defined to be 0.10 radians for rotation and within five pixels in **X** and **Y** for translation), the next virtual image from the morph sequence was used. The process terminated when all the images generated in the morph had been utilized.

The movement in depth was controlled such that the size of the object (as defined by the number of pixels inside the contour) was within 3% of the template. From this point, the final movement in depth (the approach) was done using the offset information collected in the learning stage. A simple example of positioning the manipulator in terms of rotation of the end-effector based on the synthetic images generated in Fig. 7 is shown in Fig. 8. Here, the images of the object observed by the manipulator during the pose alignment process are shown interleaved with the synthetic images generated during the morph. Each synthetic image (shown as the object contour and numbered 1 through 5) corresponds to a view from a desired position which the manipulator needs to attain. The real views of the object, obtained by using a synthetic image as a subgoal are superscripted with the same number as the corresponding synthetic sub-goal and alphabetically subscripted to show their ordering by time. In this example, the eye-in-hand system is required to rotate clockwise to reach the desired alignment. However, due to the unknown calibration transform, ${}^{R}_{C}T$, it initially rotates counter-clockwise (view 1_a and view 1_b). As the calibration is updated the manipulator moves correctly to reach

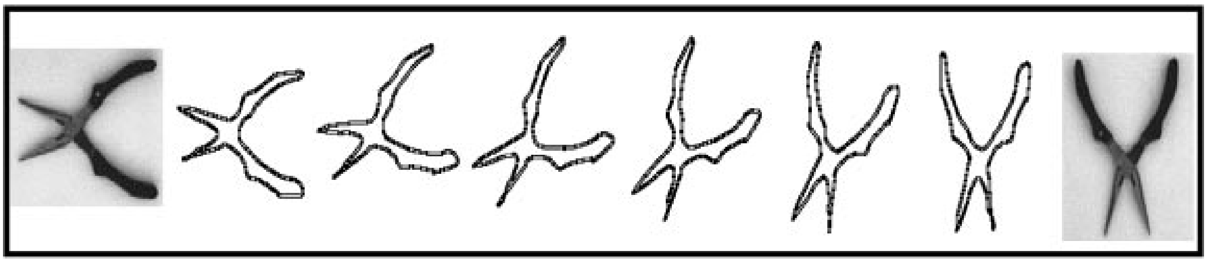


Figure 7. Contour morph sequence for the pliers. The first and the last images show the input and the desired orientations. All the intermediate images are synthesized during the morph.

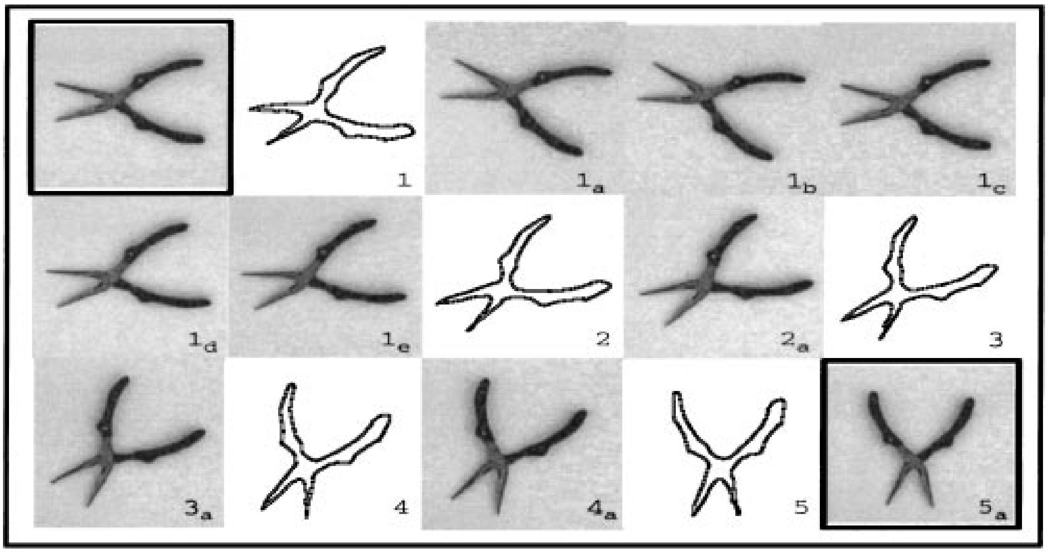


Figure 8. Pose (rotational) alignment for a pair of pliers: The views obtained from the eye-in-hand system are interleaved with the virtual images generated by the morph. Initial and target frames are shown in black borders. The robot initially starts moving the wrong way due to unknown transformation of the camera. As it moves, it estimates the transform and gets on track.

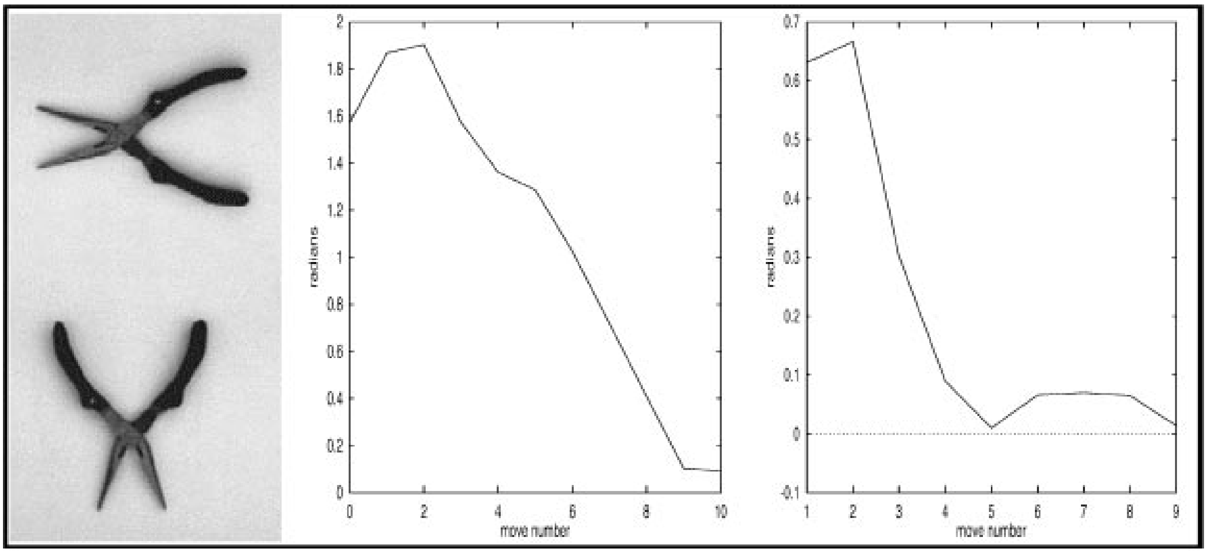


Figure 9. Experimental results for a pair of pliers. The images on the left indicate the starting view (top) and the desired view (bottom). The first graph shows the decrease in the angular error, with respect to the final desired pose. The second graph shows the change in the angular error as each new sub-goal is introduced to guide the manipulator to the desired alignment position.

the sub-goal in terms of the aforementioned predefined threshold (views 1_c through 1_e). At this point, the next synthetic image is introduced. The effect of the online estimating of the calibration on convergence can be noticed, as from this stage onwards (images 2, 3, 4, and 5), only a single move is required to reach the corresponding sub-goals.

In Fig. 9, two plots describing the above experiment are shown. The start and the target views are shown, respectively in the top left and bottom left images of this figure. The first plot describes the error from the desired rotational alignment as a function of the moves made by the manipulator. The increase in the rotational error, as depicted in the initial part of the process is due to the absence of calibration information. As described in the previous paragraph, the motion executed by the manipulator, is guided by the synthetic images generated from the morph. In the second plot, this is profiled by plotting the error between each real image and the corresponding sub-goal (synthetic image). As the manipulator achieves a sub-goal, a decrease in the error can be observed for the corresponding move number (modulo the threshold). At this point, a new synthetic image is introduced as a sub-goal, causing the error to increase and giving the plot a non-monotonic fluctuating character.

Results from a translational alignment experiment with a wedge are shown in Fig. 10. Since there is no rotation in this case, the errors are measured in pixels (image coordinates) and Cartesian displacement (world coordinates). The changes in the alignment error, in the morph plane (left) and world coordinates (right) are depicted in the top row. The lower left graph depicts the changes in the error as the translation converges to each sub-goal and a new synthetic image is introduced to define the next sub-goal. The bottom right graph shows the trajectory of the gripper from the start till the end of the experiment. It may be noted from this plot, how the lack of calibration estimation causes the robot to move in the wrong direction initially.

In Figs. 11–14, we respectively present the results of grasping a prism, vacuum cleaner, tape dispenser, and a stapler. Each of these objects was arbitrarily placed in the workspace of the manipulator. The results for the prism are presented in Fig. 11. In this case attaining the desired position required the manipulator to execute translational motion only (no rotational misalignment). Furthermore, the uniform nature of the object's shape allowed a straightforward definition of the grasp position around the center of the prism. However, for this experiment, the gripper was placed *closer* to the object in the test case than in the training case, where the object view annotated with the *distance to grasp* information was captured. Therefore, the manipulator had to reach a point *higher* than where it started, to use the distance-to-grasp information and execute the grasp. The change in the translational error in the morph plane is shown in part (b). The oscillations in the error

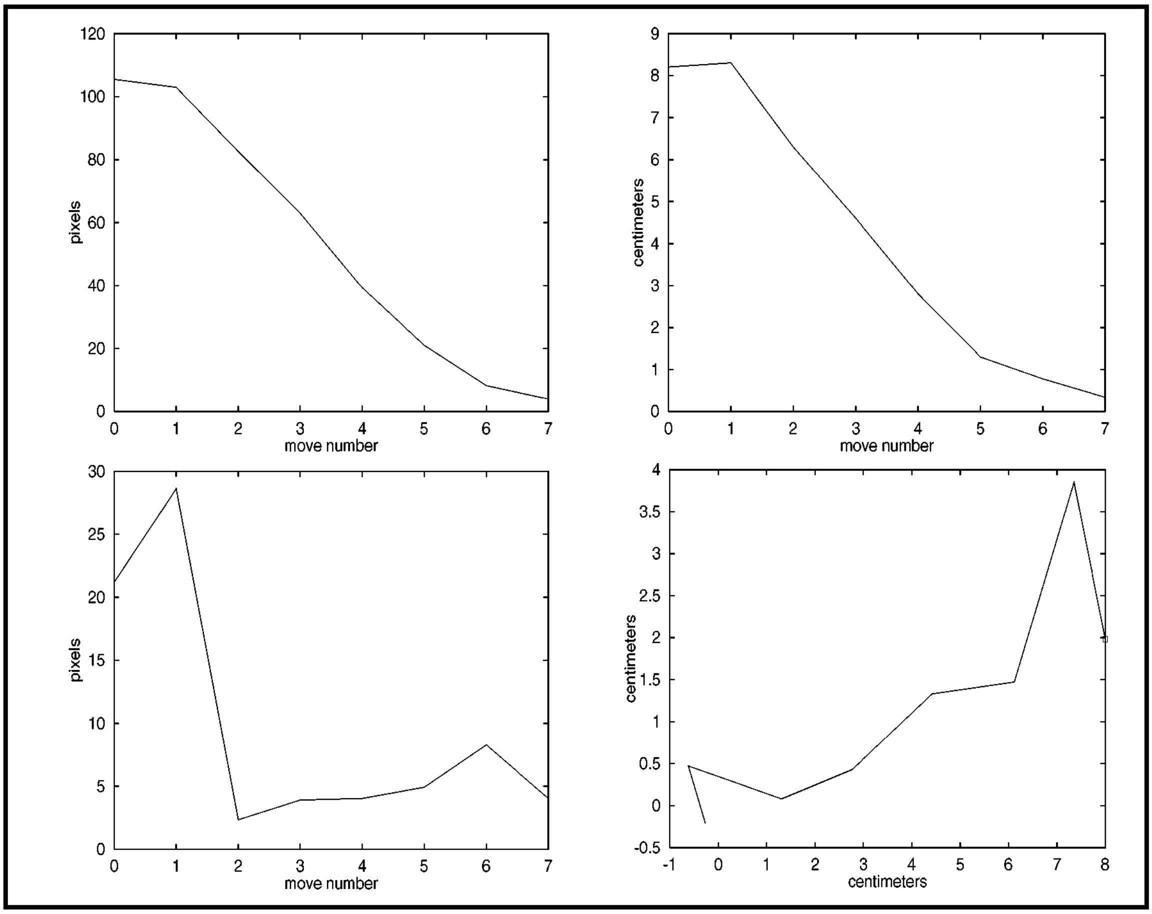


Figure 10. Translational pose alignment with a wedge. In the top row, the error from the desired pose in the morph plane is plotted on the left while the error from the desired pose in world coordinates is plotted on the right. The pose error between each virtual image (sub-goal) and its corresponding real image is shown in the left graph in the bottom row. In the right graph the planar trajectory of the gripper is depicted in world coordinates.

plot caused due to the introduction of a new synthetic images can be seen in part (c), where the error is plotted with respect to each sub-goal. The alignment of the manipulator to the object, in world coordinates is shown in part (d). In part (e), the trajectory of the gripper, as it executes the grasp is shown. Note, that the distance-to-grasp information was computed from a point that was over two feet above the object. The final approach was therefore done in two moves (move number 5 and 6) to circumvent controller thresholds on the maximum distance the end-effector could be displaced in a single move. The trajectory of the manipulator is shown in part (f).

In Fig. 12 plots describing the grasping of a vacuum cleaner are presented. For this object, the grasp position was defined on the narrow handle (and was therefore offset from the centroid of the image). Attaining the grasp position required the eye-in-hand system to position itself in terms of both translation and rotation. Plots (b), (c), and (d) profile the translational alignment process. The rotational alignment is depicted in plot (e) (error in the morph plane) and plot (f) (error in world coordinates). The reader may note the overrotation of the end-effector in move 2. Furthermore, since the grasp point is offset from the centroid of the object, rotational correction induces errors in translational alignment, that need to be rectified after the rotational alignment is complete. This effect can be seen in graph (d) where 19 moves are required to complete the translational alignment. In this experiment, the height of the end-effector above the object was the same for both the training and test cases. No height adjustment was therefore required and the stored distance-to-grasp information could be directly used to

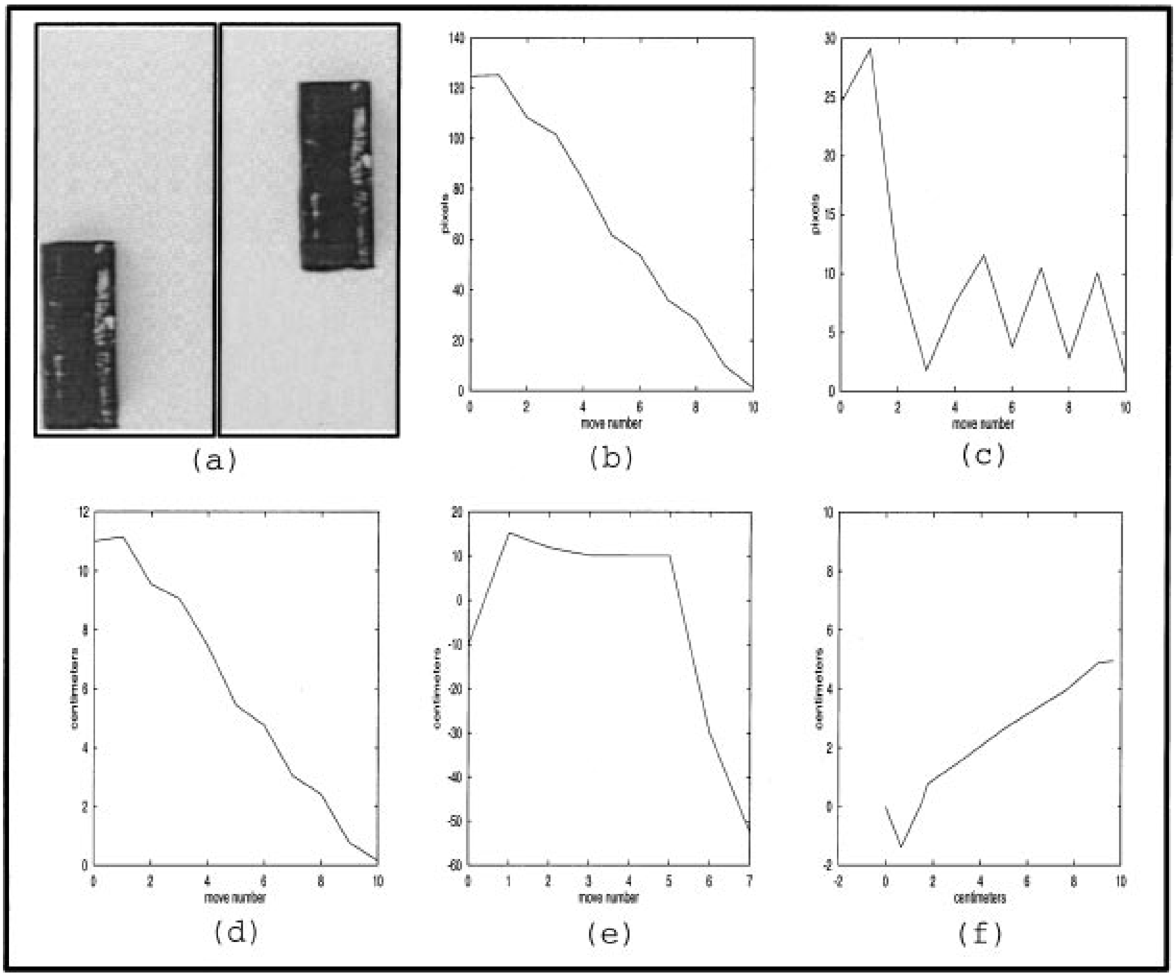


Figure 11. Results of translational pose alignment and grasping of a prism: (a) View at start (left) and desired view (right). (b) Error in the morph plane from the final position. (c) Error in the morph plane with respect to each virtual image used as a sub-goal. (d) Error in the world coordinates. (e) Movement in depth. (f) Trajectory during the alignment.

plan the final approach. This is depicted by the smooth monotonically decreasing graph in part (g). The overall trajectory of the manipulator in world coordinates is shown in the final graph.

Object 4 (the tape dispenser) considered by us was interesting because, unlike the other shapes, it could fit the gripper exactly about its middle only. Due to the curved nature of its surface, even very small deviations from this position (about 0.5 cm), rendered the object ungraspable. Using the proposed method, the object was successfully grasped in every experimental run. Results from one such run are shown in Fig. 13. Since the grasp position was very close to the object center, unlike the previous example, rotational alignment did not induce translational errors. In part (h) of this figure, the graph depicts the movements of the end effector as it positions itself to execute the grasp. This corresponds to the images shown in Fig. 4.

The results of grasping the final object (Stapler) are depicted in Fig. 14. In this example, one can note the distortions caused due to the viewing angle in the first image (part (a)). However, this distortions did not effect the recognition of the object. Since the grasp position for this object is at its top (offset from the center), rotational alignment, when following translational alignment, induced errors that had to be rectified by translating the end-effector. This effect caused the

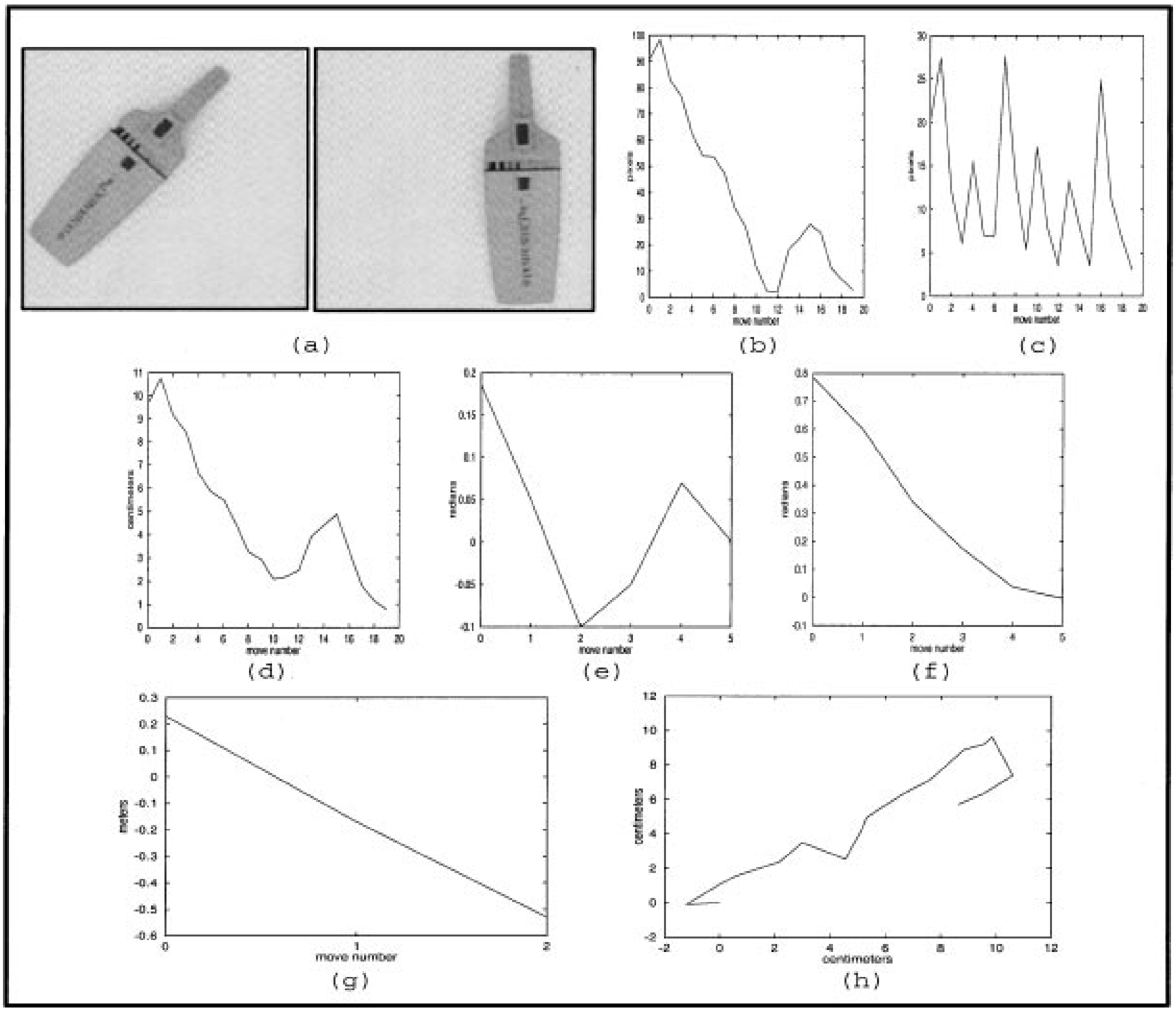


Figure 12. Grasping a vacuum cleaner: (a) View at start (left) and desired view (right). (b) Error in the morph plane from the final position during translational alignment. (c) Error in the morph plane with respect to each virtual image used as a subgoal for translational alignment. (d) Translational error in the world coordinates. (e) Rotational error in the morph plane. (f) Rotational trajectory of the robot in the world coordinates. (g) Movement in depth. (h) End-effector trajectory during the alignment.

spike in the translational error plots (graphs (b), (c), and (d)) at move number 14. The effects of this correction can also be seen in graph (i), where the trajectory of the end-effector is plotted.

5. Conclusions and Future Work

We have presented the theory underlying the control of vision-based robotic tasks involving hand-eye coordination using synthetic images created by shape morphing. The basic idea of the approach lies in mapping the virtual changes in the morph plane to the real motion of the manipulator. Specifically, the morph is used in three ways: to recognize objects from a known set, to specify a series of sub-goals along the trajectory, and to calculate the image plane error for servoing to those sub-goals. The practical application of this framework has been demonstrated by using it to control a PUMA 560 eye-in-hand system, to grasp objects of varying shape complexity.

A fundamental contribution of this approach is to link, both conceptually and algorithmically, the issue of object recognition with that of planning shape-based interactions with the object. With respect to contributions in visual servoing, the proposed method allows the

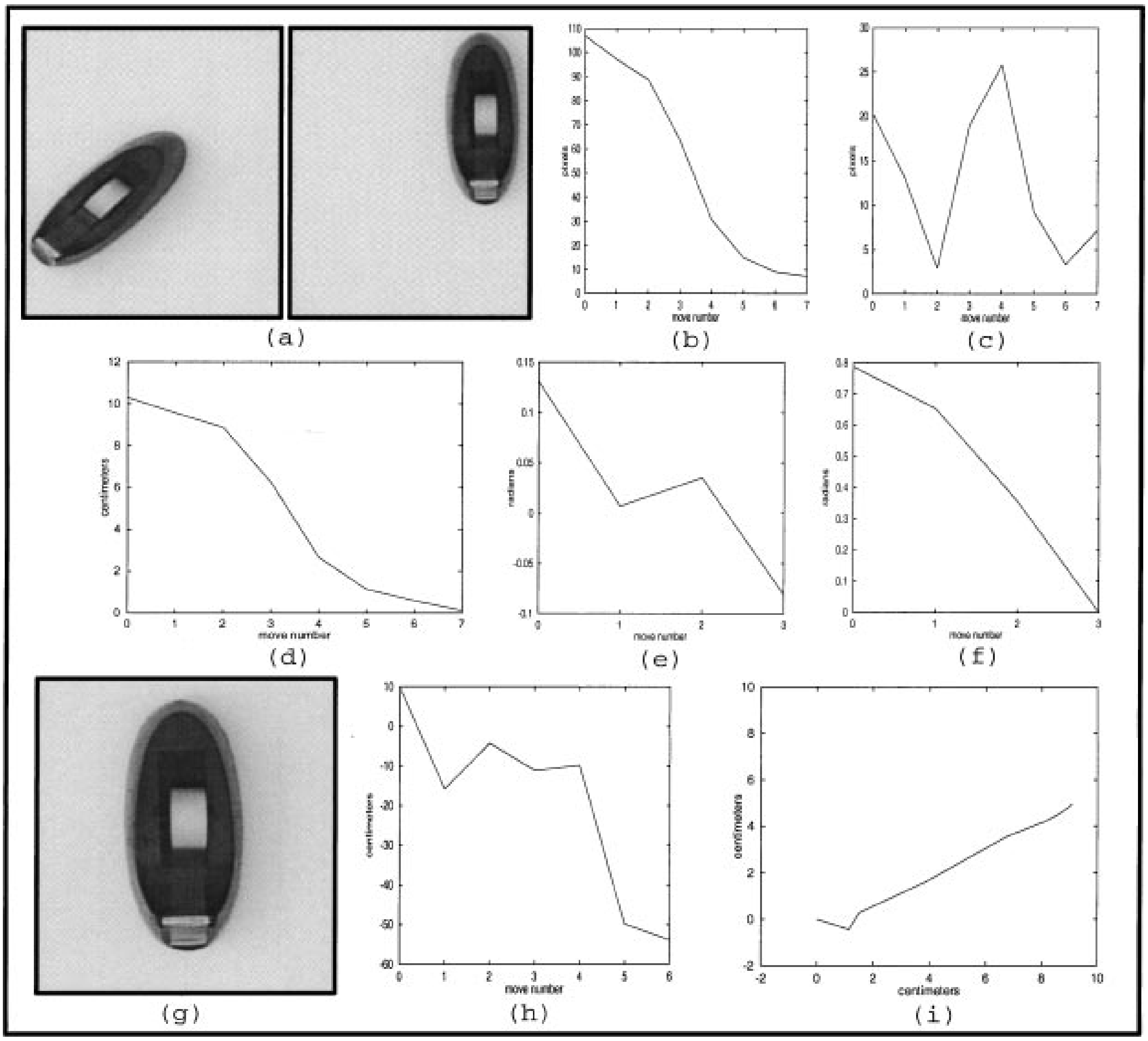


Figure 13. Results of pose alignment and grasping of a tape dispenser: (a) View at start (left) and desired position in the image plane (right). (b) Translational error in the morph plane from the final position. (c) Translational error in the morph plane with respect to each virtual image used as a sub-goal. (d) Translational error in the world coordinates. (e) Rotational error in the morph plane. (f) Rotational trajectory in the world coordinates. (g) Appearance of the object at the alignment pose and height. (h) Trajectory of the end-effector in depth. (i) Trajectory during the alignment.

specification of different grasp positions for different objects and does not involve manual feature selection and correspondence. Furthermore smooth trajectories can be obtained by generating an arbitrary number of intermediate images, allowing for path planning. It should be mentioned that our method has some similarities, in the general sense, to the work of Chaumette's group (Malis et al., 1999). The differences are in the way that the computer vision is coupled with the servoing component. In our work, there is a very strong coupling that allows us to use more complicated and realistic objects. In their case (Malis et al., 1999), the objects are black/white ones with clear landmarks.

Although our work is apparently limited by dependence on planar projections of the object contour, the limitation is not overtly constraining. While the recognition accuracy of complex objects at oblique elevation angles could decrease rapidly, the morph will always produce accurate translation vectors. As the eye-in-hand system is translated to a less oblique view, the

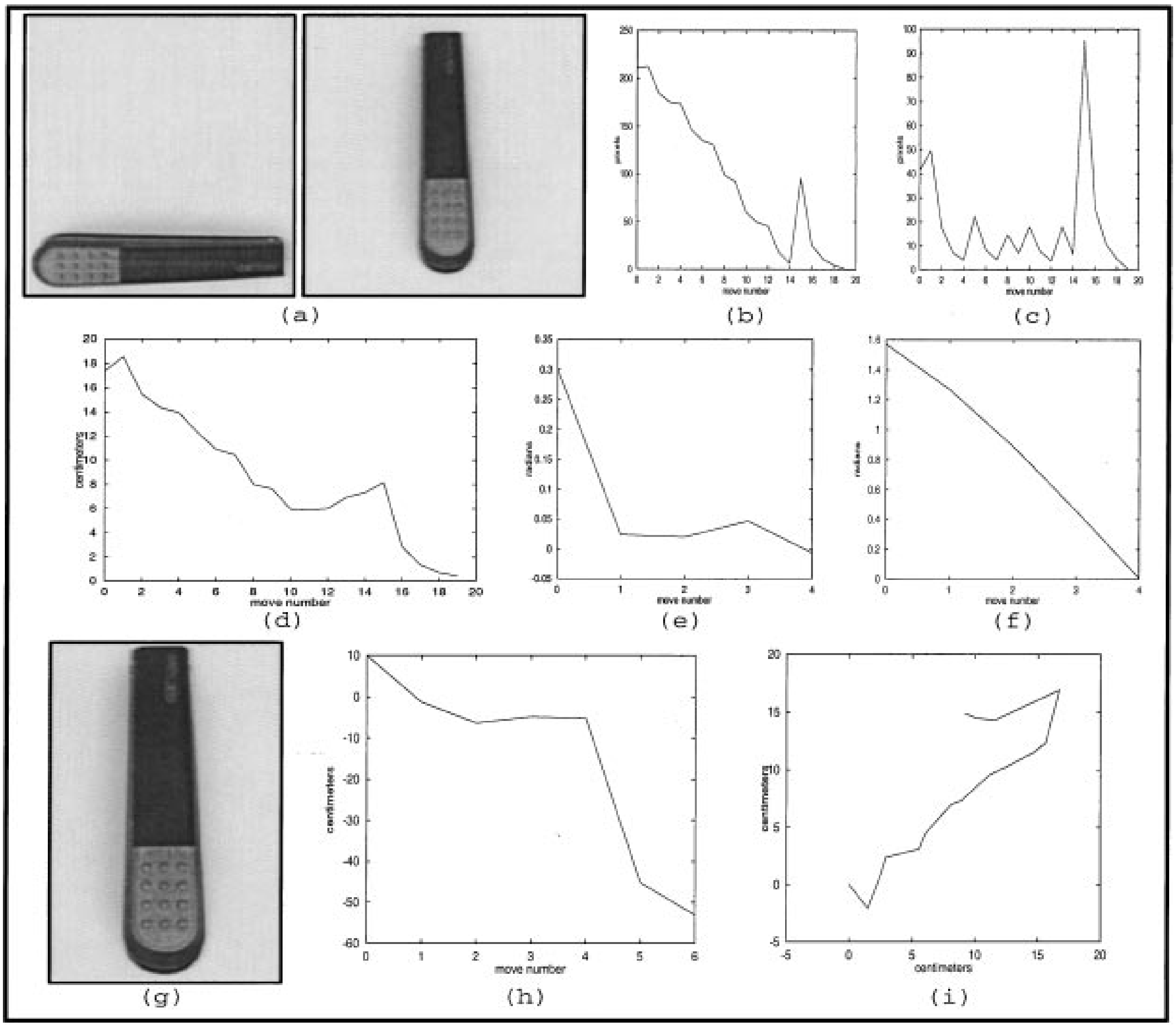


Figure 14. Results of pose alignment and grasping of a stapler: (a) View at start (left) and the alignment position (right). (b) Translational error in the morph plane from the final position. (c) Translational error in the morph plane with respect to each virtual image used as a sub-goal. (d) Translational error in the world coordinates. (e) Rotational error in the morph plane. (f) Rotational trajectory in the world coordinates. (g) Appearance of the object at the alignment pose and height. (h) Trajectory of the end-effector in depth. (i) Trajectory during the alignment.

recognition morph could be re-evaluated (as described in Section 3.2.), to correctly identify the object and, hence, the appropriate grasp position.

We believe that the ideas explored in this work may provide the basis for a fundamentally new approach to many problems in vision-based robotics. The seamless integration of object recognition and servoed trajectory generation is an important new development that provides obvious hooks for external and, eventually, integrated path planning. Although many potential applications are essentially planar and well-suited, the foremost challenge we are attempting to address is the extension to 3D grasping in a full 6-DOF formulation. One possible direction to explore in this context is to use results for generating photorealistic 3D morphs using either the fundamental matrix (Seitz and Dyer, 1996), or the trilinear tensor (Avidan and Shashua, 1997). Furthermore, in the current system a single image of an object was used as a template. A possible direction of future research would be to research the use of multiple views of an object as templates and its influence on the robustness of the recognition results. Another potential application may be motivated by our results in content-based retrieval problems. There, we used shape morphing with images of objects as templates and hand drawn outlines as the input. Extension of this idea, to visual servoing, may allow an operator to vary the approach and explore an object, by specifying (drawing) it at different orientations. A more direct application could be the extension of this system to the *programming by human demonstration framework*. In such a scenario, the visual specification of grasp point and grasp orientation would be a natural part of the paradigm.

Acknowledgments

The presentation of this paper has gained significantly from the detailed comments of the anonymous reviewers. The authors would like to extend their gratitude to them. The research of Rahul Singh on this project was supported by the National Science Foundation through Grants Nos. IRI-9410003 and IRI-9502245.

References

Allotta, B. and Colombo, C. 1990. On the use of linear camera-object interaction models in visual servoing. *IEEE Trans. Robotics and Automation*, 15(2):350–357.

Avidan, S. and Shashua, A. 1997. Novel view synthesis in tensor space. In *Computer Vision and Pattern Recognition 1997*, pp. 1034–1040.

Borst, Ch., Fischer, M., and Hirzinger, G. 1999. A fast and robust grasp planner for arbitrary 3D objects. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 3, pp. 1890–1896.

Brault, J.J. and Plamondon, R. 1993. Segmenting handwritten signatures at their perceptually important points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):953–957.

Chaumette, F. and Rives, P. 1990. Vision-based-control for robotic tasks. In *Proceedings of the IEEE International Workshop on Intelligent Motion Control*, pp. 395–400.

Chaumette, F., Rives, P., and Espiau, B. 1991. Positioning of a robot with respect to an object, Tracking it and estimating its velocity by visual servoing. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2248–2253.

Corke, P., Roberts, J., and Winstanley, G. 1998. Vision-based control for mining automation. *IEEE Robotics and Automation Magazine*, 5(4):44–49.

Feddema, J.T. and Lee, C.S.G. 1990. Adaptive image feature prediction and control for visual tracking with a hand-eye coordinated camera. *IEEE Trans. on Syst., Man, and Cybern.*, 20(5): 1172–1183.

Ferrari, C. and Canny, J. 1992. Planning optimal grasps. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 3, pp. 2290–2295.

Gangloff, J., De Mathelin, M., and Abba, G. 1999. Visual servoing of a 6-DOF manipulator for unknown 3d profile following. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 4, pp. 3236–3342.

Ghosh, B.K., Xiao, D., Xi, N., and Tarn, T.J. 1996. Calibration-free vision-based control of a robotic manipulator. In *Proceedings of the Thirty-Fourth Annual Allerton Conference on Communication, Control and Computing*, pp. 593–602.

Hager, G.D. 1995. Calibration-free visual control using projective invariance. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1009–1015.

Hashimoto, K., Ebine, T., and Kimura, H. 1992. Dynamic visual feedback control for a hand-eye manipulator. In *Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1863–1868.

Hester, R.D., Cetin, M., Kapoor, C., and Tesar, D. 1999. A criteria-based approach to grasp synthesis. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 1255–1260.

Horn, B.K.P. 1986. *Robot Vision*. MIT Press: Cambridge, MA.

Hutchinson, S., Hager, G.D., and Corke, P.I. 1996. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–670.

Janabi-Sharifi, F. and Wilson, W.J. 1998. Automatic grasp planning for visual-servo controlled robotic manipulators. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 28(5):693–711.

Joshi, R. and Sanderson, A.C. 1998. Application of feature-based multi-view servoing for lamp filament alignment. *IEEE Robotics and Automation Magazine*, 5(4):25–32.

Kuniyoshi, Y., Inaba, M., and Inoue, H. 1994. Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *IEEE Transactions on Robotics and Automation*, 10(6):799–822.

Malis, E., Chaumette, F., and Boudet, S. 1998. Positioning a coarse-calibrated camera with respect to an unknown object by 2-1/2-D visual servoing. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 1352–1359.

Malis, E., Chaumette, F., and Boudet, S. 1999. 2-1/2-D visual servoing. *IEEE Trans. Robotics and Automation*, 15(2):238–250.

Papanikolopoulos, N. and Khosla, P. 1992. Robotic visual servoing around a static target: An example of controlled active vision. In *Proceedings of the 1992 American Control Conference*, pp. 1489–1494.

Papanikolopoulos, N.P. and Khosla, P.K. 1993. Adaptive robotic visual tracking: Theory and experiments. *IEEE Trans. on Automatic Control*, 38(3):429–445.

Papanikolopoulos, N.P., Nelson, B.J., and Khosla, P.K. 1995. Six degree-of-freedom hand/eye visual tracking with uncertain parameters. *IEEE Transactions on Robotics and Automation*, 11(5):725–732.

Pavlidis, I., Singh, R., and Papanikolopoulos, N.P. 1998. On-line handwriting recognition using physics-based shape metamorphosis. *Pattern Recognition*, 31(11):1589–1600.

Perrin, D., Masoud, O., Smith, C.E., and Papanikolopoulos, N.P. 2000. Unknown object grasping using statistical pressure models. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 1054–1059.

Ray, B.K. and Ray, K.S. 1993. Determination of optimal polygon from digital curve using L_1 norm. *Pattern Recognition*, 26(4): 505–509.

Salganicoff, M., Ungar, L.H., and Bajcsy, R. 1996. Active learning for vision-based robot grasping. *Machine Learning*, 23(2–3): 251–278.

Sederberg, T.W. and Greenwood, E. 1992. A physically based approach to 2D shape blending. *Computer Graphics*, 26(2):25–34.

Seitz, S. and Dyer, C.R. 1996. View morphing. In *Computer Graphics Proceedings, SIGGRAPH96*, pp. 21–30.

Singh, R. and Papanikolopoulos, N.P. 2000. Planar shape recognition by shape morphing. *Pattern Recognition*, 33(10):1683–1699.

Smith, C.E. and Papanikolopoulos, N.P. 1996. Vision-guided robotic grasping: Issues and experiments. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3203–3208.

Smith, G., Lee, E., Goldberg, K., Bohringer, K., and Craig, J. 1999. Computing parallel-jaw grips. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 3, pp. 1897–1903.

Vikramaditya, B. and Nelson, B.J. 1999. Visually servoed micropositioning for robotic micromanipulation. *Microcomputer Applications*, 18(1):23–31.

Vincze, M. and Hager, G. 2000. *Robust Vision for Vision-Based Control of Motion*. IEEE Press: New York.

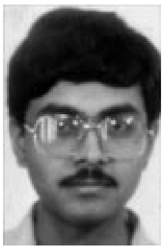
Voyles, R.M. and Khosla, P.K. 1999. Gesture-based programming: A preliminary demonstration. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 1, pp. 708–713.

Weiss, L.E. 1984. Dynamic visual servo control of robots: An adaptive image-based approach. Ph.D. Thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University.

Weiss, L., Sanderson, A., and Neuman, C. 1987. Dynamic sensor-based control of robots with visual feedback. *IEEE Journal of Robotics and Automation*, 3(5):404–417.

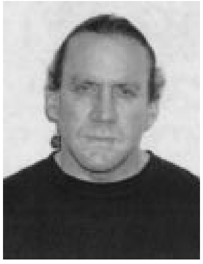
Wilson, W.J., Hulls, C.C.W., and Bell, G.S. 1996. Relative end-effector control using cartesian position based visual servoing. *IEEE Trans. Robotics and Automation*, 12(5):684–696.

Yoshimi, B.H. and Allen, P.K. 1995. Alignment using an uncalibrated camera. *IEEE Trans. on Robotics and Automation*, 11(4):516–521.



Rahul Singh received his masters degree in Computer Science, “with excellence”, from the Moscow Power Engineering Institute and the M.S. and Ph.D. degrees in Computer Science from the University of Minnesota, specializing in Computer Vision and Vision-based Robotics. Since 1998, he has been in Exelixis, a biotech company in the bay area. Both in his current capacity as group leader of the Computational Drug Discovery group, and previously as a senior research scientist and research scientist, he has been investigating problems related to molecule-molecule interactions and rapid drug discovery by using computational vision techniques to analyze molecular shape. Dr. Singh was a recipient of a fellowship for higher education from the USSR ministry of education and received the “Outstanding Scholarly Contribution Award” and the “Research Contribution Award” from the University of Minnesota. His primary research interests are in Computer Vision, Pattern Recognition, Vision-based Robotics, and computational problems in Biology and drug discovery.

Richard M. Voyles received the B.S. in Electrical Engineering from Purdue University in 1983, the M.S. in Manufacturing Systems Engineering from the Department of Mechanical Engineering at Stanford University in 1989, and the Ph.D. in robotics from the School of Computer Science at Carnegie Mellon University in 1997. He is currently an Assistant Professor in the Department of Computer Science and Engineering at the University of Minnesota and a Senior Member of the IEEE. Dr. Voyles’ industrial experience includes three years with IBM, where he was a manufacturing/test engineer working on automation projects, and one and one-half years with Intergrated Systems, Inc., where he was a research scientist working on contracted applications of real-time software. Dr. Voyles founded one company, Trident Robotics and Research, Inc., and co-founded another, both to address issues in real-time control hardware and software. He is currently on the Technical Advisory Board of Point Cloud, Inc. Dr. Voyles’ research interests are in the areas of robotics and artificial intelligence. Specifically, he is interested in the coordination of teams of robotic agents for common goals where resource constraints play an important role. He is also interested in mobile manipulation, programming robots by human demonstration, and agent-to-agent skill transfer. His interests in computer vision include extracting 3-D models of objects from a moving camera. Dr. Voyles also has expertise in sensors and sensor calibration, particularly haptic and force sensors.



David Littau received his Bachelors degree in Mechanical Engineering and Applied Mathematics from the University of California, San Diego, and his Master’s degree in Mechanical Engineering from San Diego State University. He is currently a Ph.D. candidate in Computer Science at the University of Minnesota. His research interests are Evolutionary Algorithms and Artificial Intelligence, and he is currently doing research in data mining.

Nikolaos P. Papanikolopoulos (S’88–M’93) was born in Piraeus, Greece, in 1964. He received the Diploma degree in electrical and computer engineering from the National Technical University of Athens, Athens, Greece, in 1987, the M.S.E.E. in electrical engineering from Carnegie Mellon University (CMU), Pittsburgh, PA, in 1988, and the Ph.D. in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 1992. Currently, he is an Associate Professor in the Department of Computer Science at the University of Minnesota. His research interests include robotics, computer vision, sensors for transportation applications, and control. He has authored or coauthored more than 130 journal and conference papers in the above areas (twenty nine refereed journal papers). He was finalist for the Anton Philips Award for Best Student Paper in the 1991 IEEE Robotics and Automation Conference. Futhermore, he was recipient of the Kritski fellowship in 1986 and 1987. He was a McKnight Land-Grant Professor at the University of Minnesota for the period 1995–1997 and has received the NSF Research Initiation and Early Career Development Awards. He was also awarded the Faculty Creativity Award from the University of Minnesota. Finally, he has received grants from DARPA, Sandia National Laboratories, NSF, USDOT, MN/DOT, Honeywell, and 3M.