2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Visual Servoing based on Gaussian Mixture Models

A.H. Abdul Hafez, Supreeth Achar, and C.V. Jawahar

Abstract— **In this paper we present a novel approach to robust visual servoing. This method removes the feature tracking step from a typical visual servoing algorithm. We do not need correspondences of the features for deriving the control signal. This is achieved by modeling the image features as a Mixture of Gaussians in the current as well as desired images. Using Lyapunov theory, a control signal is derived to minimize a distance function between the two Gaussian mixtures. The distance function is given in a closed form, and its gradient can be efficiently computed and used to control the system. For simplicity, we first consider the 2D motion case. Then, the general case is presented by introducing the depth distribution of the features to control the six degrees of freedom. Experiments are conducted within a simulation framework to validate our proposed method.**

I. INTRODUCTION

Visual servo control or visual servoing is the process of positioning a robot end effector with respect to an object or a set of features extracted from the object of interest. Visual information like image points, contours, and line segments are typically used as features [1], [2].The most commonly used features in the visual servoing algorithms are image points. These features could be some interest points, computed out of the gray-level invariance. In these situations, feature tracking is generally necessary to provide the required correspondences (across the images) to compute the control signal. These correspondences could be either between the current and desired camera frame or between two consecutive frames of the video.

Feature tracking in an image sequence is an important topic of research in visual servo control and general robot vision applications [3]. Fig 1 shows the role of the tracking step in classical visual servo control process. Robust extraction and real-time tracking of visual cues is vital to the performance of a visual servo control system. For practical applicability in most real-life environments, one needs to derive the control signal by robustly tracking without any fiducial markers. Many researchers have recently been investigating the tracking stage of the visual servoing process. A number of visual tracking methods, which are robust to noisy measurements and changes in environmental conditions, have been proposed. Kragic [4] presented a 2D tracking algorithm based on integration techniques which provides robust visual information to the control law. Hafez *et al.* [5] presented a planar tracking algorithm based on the probabilistic integration of edge and image point features. A similar algorithm [6] has been proposed by Pressigout and Marchand within a deterministic optimization framework. However, these algorithms still involve an expensive tracking step to provide the correspondences and other required information like pose vector or homography matrix to the control law. This tracking step is usually derived in a recursive framework. They may also require additional computations to provide robustness.

Visual servoing methods have been proposed which integrate the tracking phase into the control law (see Fig 2). In [7] Tahri and Chaumette presented a moment-based approach to visual servoing which computes the moments of a planar contour without explicitly considering point cor-

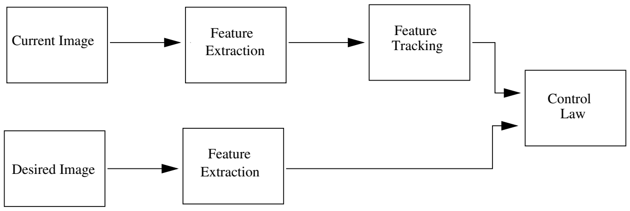

Fig. 1. Tracking step within traditional visual servo control framework. Features are tracked and compared with the desirved one to derive the control signal

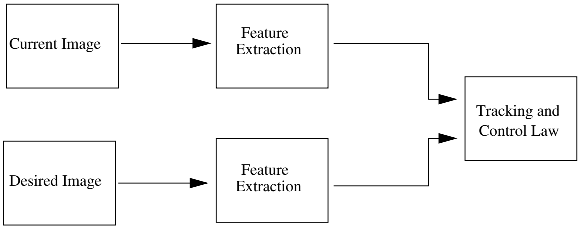

Fig. 2. Tracking step in some of the recent algorithm is integrated with the control law. (See Sec I)

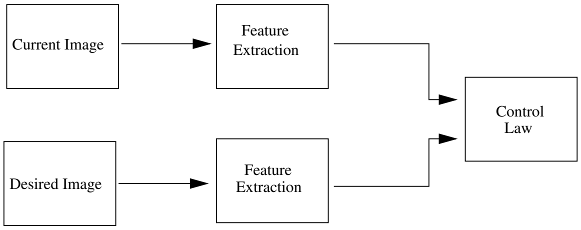

Fig. 3. Tracking step is avoided in our visual servo control algorithm. We model the feature distribution as GMM and derive the control signal based on GMMs

Authors are with Center for Visual Information technology, International Institute of information Technology, Gachibowli, Hyderabad-500032, India {hafez@research.,supreeth@students.}iiit.ac.in jawahar@iiit.ac.in

978-1-4244-1647-9/08/$25.00 ©2008 IEEE.

respondences. The translation is controlled using zeroth and first order moments while the rotation control needs higher order moments which are more sensitive to noise. In a more recent work, the same authors proposed to control rotation using a homography decomposition based method [8]. However, the method still requires a contour tracker. Recently, Kallem *et al.* [9] proposed a kernel-based visual servoing method that does not separate the tracking stage from the control stage. Instead, authors use a tracking-based kernel function to generate the control signal. However, at present, the method is limited to four degrees of freedom *i.e.* (3D translation and roll). Another attempt to unify the tracking stage with the servo control stage has been proposed by Malis and Benhimane [10]. Their method uses an efficient second order approach to homography tracking and estimation of a planar object through a sequence of images. In this case too, tracking stage is not completely avoided; but is merged into the control stage. Figure 3 shows an outline of what is presented in this paper, a visual servo control in which the tracking step has been removed completely.

In this paper we propose a novel approach to visual servoing. This method uses a set of image points extracted from the target image to compute the required control signal without the use of explicit point-to-point correspondences. The extracted features are fed directly to the control law. In our algorithm, we model the M image point features extracted from the current image and the N image point features extracted from the desired image as two separate Gaussian Mixture Models (GMMs). We assume that in the most general case, that N and M need not be equal. There is as such no correspondence information between the point sets extracted from the current and target images. Our method is based on a closed form distance function between two GMMs that was proposed in [11]. We use Lyapunov theory to compute a control signal that minimizes this distance function. First, we demonstrate the approach with a 2D motion involving three degrees of freedom. We also show that this method can be generalized to the six degree of freedom motion by introducing the depth distribution of the features.

Since we treat the feature points collectively and model them as a distribution (rather than working with individual feature points) our method is statistically robust. Hence it is suitable for applications in real world environments where features can not be detected perfectly.

II. MODELING AND PROBLEM FORMULATION

A. Notations

The pose P of the end effector of the robot arm can be represented in terms of its rigid transformation $[R, t]$ with respect to a world coordinate system. Here, R is a 3×3 rotation matrix and t is the 3×1 translation vector $t = [t_x \ t_y \ t_z]^T$. This representation of the pose is used when manipulating Gaussian mixtures. The rotation matrix R can be parameterized in terms of 3 Euler angles (α, β and γ) such that $R = R_x(\alpha)R_y(\beta)R_z(\gamma)$. When deriving the control signal it is more convenient to use the Euler angle parameterization. In this case, the camera pose P is represented by 6×1 vector $[t_x \ t_y \ t_z \ \alpha \ \beta \ \gamma]^T$.

B. Gaussian Mixture Model

The Gaussian density function $\mathcal{N}(X; \mu, \Sigma)$ represents the probability density with mean μ and variance Σ evaluated at X. A Gaussian Mixture Model or GMM is a probability distribution which is a convex combination of Gaussian distributions $f(x) = \sum_{i=1}^{i=n} \alpha_i \mathcal{N}(X; \mu_i, \Sigma_i)$ where α is a sequence of n non negative weights that sum to one. Efficient algorithms exist for fitting a GMM to a set of samples using an Expectation Maximization formulation. We use GMMs to model distributions of feature points in the image space or Cartesian space. Each feature point contributes one component Gaussian to the model with mean equal to its observed position and a variance that represents the uncertainty in its position.

C. Similarity between two Gaussian Mixture Model

Many distance measures such as the Kullback-Leibler divergence and the L_2 norm can be used as a distance measure between two Gaussian mixtures. We use the L_2 distance to quantify the similarity between Gaussian mixtures. The reason for that is the availability of a closed form expression for the L_2 distance between GMMs, which allows for implementation with lower time and space complexity [11].

The GMM of the N feature points in the desired view is denoted by $g(X) = \sum_{j=1}^{N} \beta_j \mathcal{N}(X; \nu_j, \Gamma_j)$ and that of the M feature points in the current view is denoted by $f(X, P) = \sum_{i=1}^{M} \alpha_i \mathcal{N}(X; R\mu_i + t, R\Sigma_i R^T)$. Changing manipulator pose applies a rigid transformation to the distribution $f(X, P)$ while $g(X)$ remains fixed. As the current pose approaches the target pose, the overlap between $g(X)$ and $f(X, P)$ will increase or equivalently the distance between them will decrease. Thus, the visual servoing process can be viewed as one that minimizes a distance measure $d(f(X, P), g(X))$ between $g(X)$ and $f(X)$ by changing the manipulator pose P. Since we are using the L_2 norm the distance $d(f, g) = \int [f(X, P) - g(X)]^2 \, dX$. This can be expanded and written as

$$d(f, g) = \int (f^2(X, P) - 2f(X, P)g(X) + g^2(X))dX. \quad (1)$$

D. Uncertainty Propagation from Image to 3D Space

This subsection describes how a Gaussian distribution in the 2D image space can be projected into 3D space as shown in Fig 5. Consider the image point $x = (u, v)$, which is a projection of the 3D point X. The measurement x of this image point can be corrupted by noise and errors. This corruption can be represented by a Gaussian distribution with zero mean and variance Σ_x. This results in a random variable with probability distribution [12], [13] given by

$$p(x \mid X) = \frac{1}{2\pi|\Sigma_x|^{1/2}} \exp[(-\frac{1}{2}(x - KX)^T \Sigma_x^{-1} (x - KX))]. \quad (2)$$

Given a probability distribution $p(Z) = \mathcal{N}(Z; \bar{Z}, \sigma_Z)$ of the depth of X with a mean $\bar{Z}$ and variance σ_Z, the image

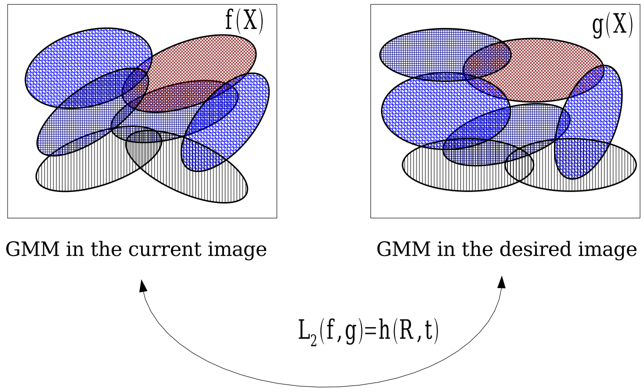

Fig. 4. GMMs in image space and the distance between them.

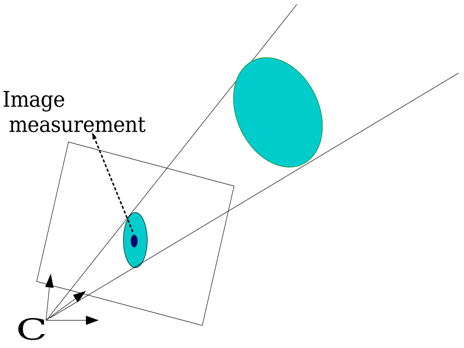

Fig. 5. Uncertainty projection from image to the Cartesian space.

measurements can be back-projected to the Cartesian space using the function F^{-1}, the inverse of the back-projection function of the camera. A probability density function of the 3D point X that corresponds to the image point measurement x is obtained. This function is $p(X \mid x) = \mathcal{N}(X; \bar{X}, \Sigma_X)$ and the parameters $\bar{X}$ and Σ_X are computed as follows [12]

$$\bar{X} = [\bar{Z}\bar{u}, \bar{Z}\bar{v}, \bar{Z}]^T, \quad \Sigma_X^{-1} = J_F^T \begin{pmatrix} \Sigma_x^{-1} & 0 \\ 0 & \sigma_Z^{-1} \end{pmatrix} J_F \ . \tag{3}$$

Here, the matrix J_F is the Jacobian of the inverse of the back-projection function [12] and defined as

$$J_F = \frac{\partial F^{-1}}{\partial X}\Big|_{\bar{X}} = \begin{pmatrix} 1/\bar{Z} & 0 & -\bar{u}/\bar{Z} \\ 0 & 1/\bar{Z} & -\bar{v}/\bar{Z} \\ 0 & 0 & 1 \end{pmatrix}. \tag{4}$$

The resulting distribution of the 3D points coordinates is used in the GMM to perform the visual servo control.

III. GMM FOR VISUAL SERVOING

A. The Design of Locally Stable Control Law

In this section we adopt a set of assumptions without the loss of generality. The visual servoing system is assumed to be an eye-in-hand visual servoing system. In other words, we have a moving camera and a stationary object. Further, we assume that the control loop is an indirect visual servoing control loop [14]. Consequently, the control input u to the robot controller is given as the screw velocity of the camera and define as

$$u = \frac{\partial P}{\partial t}. \tag{5}$$

Here the camera pose P is represented as a 6×1 vector. The control signal u is also a 6×1 vector.

Our cost function $\mathcal{C}$ is the L_2 distance between the Gaussian mixtures $f(X,P)$ and $g(X)$

$$\mathcal{C} = \int [f(X,P) - g(X)]^2 \, dX, \tag{6}$$

$$\mathcal{C} = \int \left[f^2(X,P) - 2f(X,P)g(X) + g^2(X)\right] dX, \tag{7}$$

where $f(X,P)$ represents a mixture of Gaussians in the current frame. It can be explicitly written as a function of the relative pose between the current camera frame and the desired one.

$$f(X,P) = \sum_{i=1}^{M} \alpha_i \mathcal{N}(X; R\mu_i + t, R\Sigma_i R^T). \tag{8}$$

Here, R and t are the rotation and translation between the current frame and the desired frame, and M is the number of point features extracted from the current image. Similarly,

$$g(X) = \sum_{j=1}^{N} \beta_j \mathcal{N}(X; \nu_j, \Gamma_j) \tag{9}$$

represents a mixture of Gaussians in the desired image. The number of image features extracted from the desired image is N where $N \neq M$ in the general case. In addition, in the desired image the rotation is $R = I$ and $t = \mathbf{0}$.

Consider the following Lyapunov candidate function

$$\mathcal{L} = \frac{1}{2}(\mathcal{C} - \mathcal{C}_0)^2, \tag{10}$$

where $\mathcal{C}_0$ is the value of the cost function $\mathcal{C}$ evaluated at the desired position. To produce a control signal that stabilizes the servoing system write

$$\frac{\partial \mathcal{L}}{\partial t} = (\mathcal{C} - \mathcal{C}_0)\frac{\partial \mathcal{C}}{\partial P}\frac{\partial P}{\partial t}, \tag{11}$$

We select the control signal u such that

$$u = -(\mathcal{C} - \mathcal{C}_0)\nabla_P(\mathcal{C}), \tag{12}$$

where $\nabla_P(\mathcal{C}) = \frac{\partial \mathcal{C}}{\partial P}$. Substituting (12) and (5) in (11), we get the derivative of the candidate Lyapunov function as

$$\frac{\partial \mathcal{L}}{\partial t} = -(\mathcal{C} - \mathcal{C}_0)^2 \parallel \nabla_P(\mathcal{C}) \parallel^2 . \tag{13}$$

One may note that the candidate Lyapunov function is positive definite in the pose variable P and its derivative function is negative semi-definite. This proves the local stability and convergence of the control law.

In the following subsections we show the details of the gradient computation $\frac{\partial \mathcal{C}}{\partial P} = \nabla_P(\mathcal{C})$ and the derivative of the visual servoing control law in case of planar motion where the rotation and translation form a 2D transformation, then generalize it to the six degree of freedom case where we assume a depth distribution is given. This is done by propagating the distribution into 3D space as explained in section II-D.

B. Gradient Computation

The distribution $g(X)$ in (7) does not depend on the transformation parameters R and t. Furthermore, the L_2 norm is an invariant under rigid transformation. Consequently, $\int f^2(X, [R, t]) = f^2(X[I, 0])$. This implies that the first and last terms of the cost function can be dropped and only the cross term needs to be considered. Therefore (7) can be rewritten as

$$\mathcal{C} = \int [f(X, P)g(X)]\, dX \tag{14}$$

$$\mathcal{C}(P) = \sum_{i=1}^{M} \sum_{j=1}^{N} \alpha_i \beta_j \mathcal{N}(\mathbf{0}; R\mu_i + t - \nu_j, R\Sigma_i R^T + \Gamma_j) \tag{15}$$

The cost function $\mathcal{C}(P)$ can be differentiated with respect to the pose parameters to obtain the gradient function $\nabla_P(\mathcal{C})$.

$$\nabla_P(\mathcal{C}) = \sum_{i=1}^{M} \sum_{j=1}^{N} \alpha_i \beta_j \nabla_P \left(\mathcal{N}(\mathbf{0}; R\mu_i + t - \nu_j, R\Sigma_i R^T + \Gamma_j) \right) \tag{16}$$

We omit the development details of the vector $\nabla(\mathcal{C})$ as it is fairly straightforward but lengthy.

We consider here two cases of eye-in-hand visual servoing. First, the camera is mounted on a planar robot end-effector that can move in x-y directions and rotate in the x-y plane, about the z axis. In this case, the control signal u is the 3-vector $\dot{P} = [\dot{t}_x \; \dot{t}_y \; \dot{\gamma}]^T$. The problem is reduced to 2D image rigid transformation; The matrix R is a 2D rotation matrix and t is a 2D translation. The Gaussian Mixtures are defined in the 2D image space by directly using the image measurements.

The second is the six degree of freedom case where we control the robot end-effector in 3D space. In other words, the robot is able to move in the three directions x,y, and z as well as to rotate about the three angles α, β, and γ. The control signal is the velocity screw $\dot{P} = [\dot{t}_x \; \dot{t}_y \; \dot{t}_z \; \dot{\alpha} \; \dot{\beta} \; \dot{\gamma}]^T$. In this case the, R is a 3D rotation matrix while t is a 3D translation vector. To propagate the uncertainty given by the image measurements from the image space to the 3D space, we assume that a depth distribution is available. Then, compute the 3D distribution as presented in Sec II-D.

IV. RESULTS

The proposed method for visual servoing using Gaussian mixture models is evaluated using a simulation framework. The validation process is done in two cases: (i) planar robot i.e. 2D motion with two directional translation and rotation about an axis that is perpendicular to the scene plane, and (ii) six degree of freedom motion that contains 3D translation as well as 3D rotation. The camera is modeled as a pinhole perspective camera with focal length and principle point given by the following internal camera parameter matrix

$$K = \begin{bmatrix} 600 & 0 & 320 \\ 0 & 600 & 240 \\ 0 & 0 & 1 \end{bmatrix}$$

The target object is a set of 3D points that are randomly drawn from the Cartesian space and projected onto the image plane using the projection model.

Our results include image trajectories of feature points which seems to imply that feature correspondences are being made. However it should be noted that these tracks are generated for visualization purposes only, the actual servoing algorithm does not make use of correspondences or a tracking step.

A. Planar Robot (2D Motion)

In the planar motion case, the task to be performed is 2D translation along the axes x-y and rotation about the z axis i.e. parallel to the camera optical axis. The control signal

$$u = \begin{bmatrix} \dot{t}_x \\ \dot{t}_y \\ \dot{\gamma} \end{bmatrix} = -(\mathcal{C} - \mathcal{C}_0)\nabla_P(\mathcal{C})$$

is computed by employing the control law derived using the Lyapunov theory for non-linear control systems. The computation is summarized by computing the gradient vector $\nabla_P(\mathcal{C}) = \frac{\partial \mathcal{C}}{\partial P}$ from (16) and substituting it in the control law given in (5). Then moving the camera based on the computed control signal and repeating this process until convergence.

Results from this visual serving task are illustrated in Figures 6-9. Fig. 6 and Fig. 7 show the control signals during the servoing process with respect to time. Translational velocity is given in Fig. 6 while the rotational velocity is given in Fig. 7. The simulation framework considers a fixed time interval and shows the result per iteration.

The image trajectory is shown in Fig. 8. The desired image point positions are indicated by red colored × marks. The camera trajectory is almost a straight line as shown in Fig. 9. However, the analytical study of the camera and image trajectory needs to be done to discover the coupling properties of the proposed control law between translation and rotation. The starting pose is

$$[R, t] = \begin{bmatrix} 0.9239 & 0.3827 & 0.75 \\ -0.3827 & 0.9239 & 0.50 \end{bmatrix},$$

while the destination pose is $[R, t] = [I, 0]$. The final camera pose at termination was

$$[R, t] = \begin{bmatrix} 0.9999 & 0.0160 & 0.0043 \\ -0.0160 & 0.9999 & 0.0049 \end{bmatrix}.$$

This yields a convergence error as $t_x = -0.0043$ meter, $t_y = -\ 0.0049$ meter, and $\gamma = 0.0162$ radians.

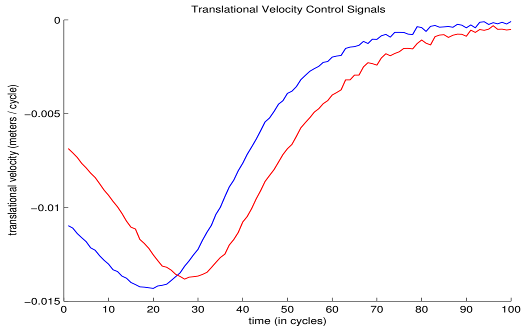

Fig. 6. The translation velocity for the planar motion task performed using the proposed algorithm.

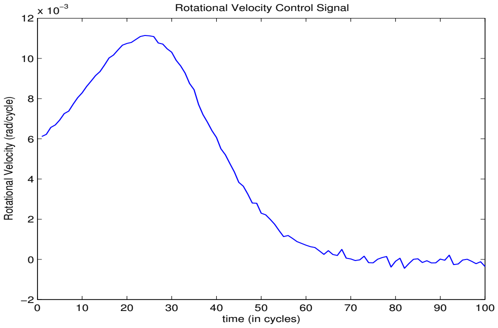

Fig. 7. The rotational velocity for the planar motion task performed using the proposed algorithm.

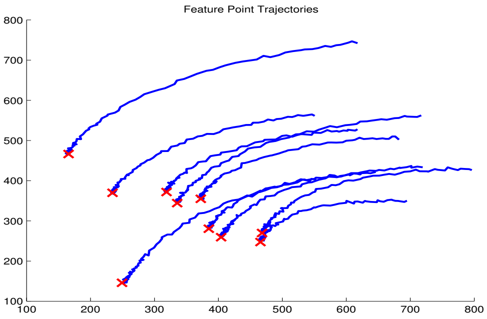

Fig. 8. Image trajectories for the planar motion task performed using the proposed algorithm. The desired position of each feature is marked with a red cross

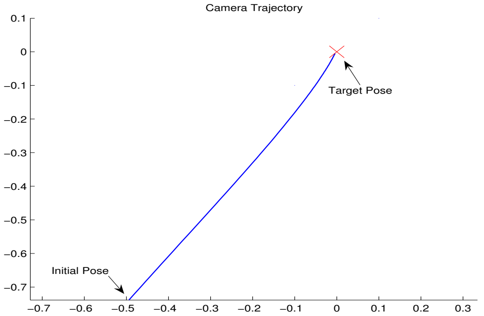

Fig. 9. Camera trajectory for the planar motion task performed using the proposed algorithm.

B. *Motion with Six Degrees of Freedom*

Here we assume that the robot is able to move along six degrees of freedom *i.e.* 3D rotation about the x, y and z axes and 3D translation along the same three axes. The control signal in this case is

$$u = V = \left[\begin{array}{cccccc} \dot{t}_x & \dot{t}_y & \dot{t}_z & \dot{\alpha} & \dot{\beta} & \dot{\gamma} \end{array}\right]^T = -(\mathcal{C} - \mathcal{C}_0)\nabla_P(\mathcal{C}).$$

The gradient $\nabla_P(\mathcal{C})$ is computed as given in (16) but with respect to 3D rotation matrix R and 3D translation vector t. In addition the distance function $\nabla_P(\mathcal{C})$ is the similarity between two 3D GMMs obtained by projection of the uncertainty from the image space to the Cartesian space using the depth distribution as shown in Sec II-D.

Results from this visual servoing task are illustrated in Figures 10-13. Fig. 10 and Fig. 11 show the control signals during the servoing process with respect to time. Translational velocity is given in Fig. 10 while the rotational velocity is given in Fig. 11. The simulation framework considers a fixed time interval and shows the result per iteration. The image trajectory is shown in Fig. 12. The desired image point positions are indicated by the red colored × marks. The camera trajectory is shown in Fig. 13.

The starting pose is

$$[R, t] = \left[\begin{array}{cccc} 1.0000 & 0 & 0 & 0.500 \\ 0 & 0.9239 & -0.3827 & 0.750 \\ 0 & 0.3827 & 0.9239 & 1.000 \end{array}\right],$$

while the destination pose is $[R, t] = [I, 0]$. The final camera pose at termination was

$$[R, t] = \left[\begin{array}{cccc} 1.0000 & -0.0034 & -0.0047 & 0.0081 \\ 0.0034 & 1.0000 & -0.0011 & 0.0001 \\ 0.0047 & 0.0011 & 1.0000 & -0.0034 \end{array}\right].$$

This yields a translational convergence error as $t_x = 0.0081$ meter, $t_y = 0.0001$ meter, $t_z = -0.0034$ meter and rotational convergence error $\alpha = 0.0011$ rad, $\beta = -0.0047$ rad, and $\gamma = 0.0034$ rad.

V. CONCLUSIONS AND FUTURE WORK

We presented a visual servoing controller that uses Gaussian Mixture models to collectively represent the image point features in the current and desired images. This modeling has removed the need for a feature tracking and matching step from the servoing process. Servoing is achieved by minimizing a distance function between the current and desired Gaussian mixture models instead of minimizing a direct error function between the corresponding features. This method is directly applicable to the case of 2D motion of a planar robot with 3 degrees of freedom. In addition, it can be extended to the six degrees of freedom by introducing the depth distribution/map of the scene.

We are presently working on exploring the statistical robustness of GMMs in the servoing process. The robustness

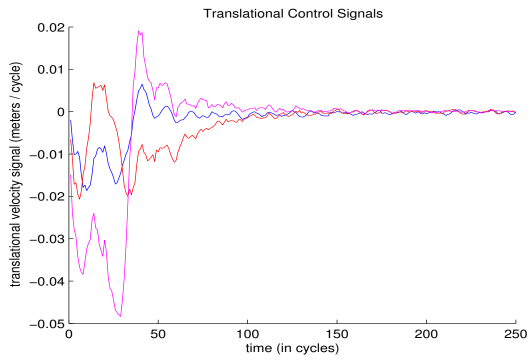

Fig. 10. The translation velocity for the 3D motion task performed using the proposed algorithm.

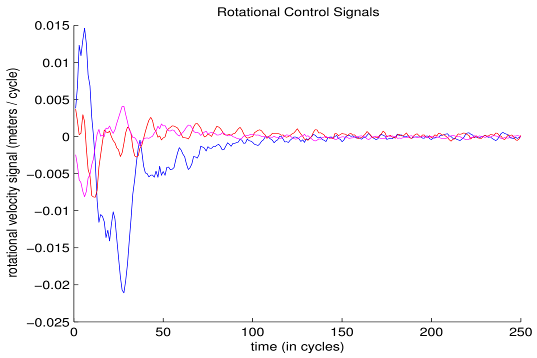

Fig. 11. The rotational velocity for the 3D motion task performed using the proposed algorithm.

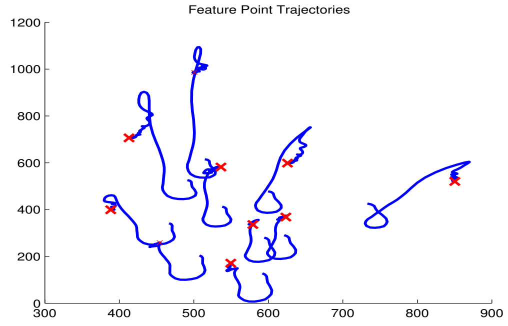

Fig. 12. Image trajectories for the 3D motion task performed using the proposed algorithm. The desired position of each feature is marked with a red cross

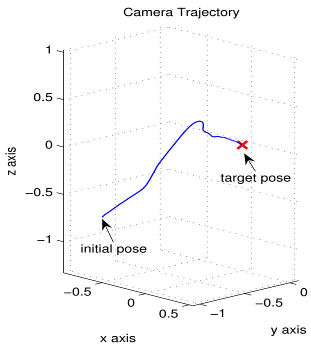

Fig. 13. Camera trajectory for the 3D motion task performed using the proposed algorithm.

of the algorithm with respect to the depth distribution is one important issue to be considered in future work. This is in addition to the issue of coupling/decoupling between rotation and translation.

References

[1] F. Chaumette and S. Hutchinson, “Visual servo control, part I: Basic approaches,” *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, December 2006.

[2] F. Chaumette and S. Hutchinson, “Visual servo control, part II: Advanced approaches,” *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, March 2007.

[3] E. Marchand and F. Chaumette, “Feature tracking for visual servoing purposes,” *Robotics and Autonomous Systems*, vol. 52, no. 1, pp. 53–70, July 2005.

[4] D. Kragic, “Visual servoing for manipulation: Robustness and integration issues,” Ph.D. dissertation, Royal Institute of Technology, Sweden, 2001.

[5] A. H. Abdul Hafez, U. Visesh Kumar, and C. V. Jawahar, “Combining texture and edge planar trackers based on a local quality metric,” in *IEEE Int. Conf. on Robotics and Automation, ICRA’07*, Roma, Italia, April 2007.

[6] M. Pressigout and E. Marchand, “Real-time planar structure tracking for visual servoing: a contour and texture approach,” in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS’05*, vol. 2, Edmonton, Canada, August 2005, pp. 1701–1706.

[7] O. Tahri and F. Chaumette, “Point-based and region-based image moments for visual servoing of planar objects,” *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1116–1127, December 2005.

[8] P. Li, O. Tahri, and F. Chaumette, “A shape tracking algorithm for visual servoing,” in *IEEE Int. Conf. on Robotics and Automation, ICRA’05*, Barcelona, Spain, April 2005, pp. 2858–2863.

[9] V. Kallen, M. Dewan, J. Swensoen, G. Hager, and N. Cowan, “Kernel-based visual servoing,” in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS’07*, San Diego, USA, 2007.

[10] E. Malis and S. Benhimane, “A unified approach to visual tracking and servoing,” *Robotics and Autonomous Systems*, vol. 52, no. 1, pp. 39–52, July 2005.

[11] B. Jian and B. Vemuri, “A robust algorithm for point set registration using mixtures of guassians,” in *International Conference on Computer Vision, ICCV’05*, Beijing, China, 2005.

[12] G. Flandin and F. Chaumette, “Visual data fusion: Application to object localization and exploration,” INRIA, Renne, France, Tech. Rep. INRIA/RR-4168-FR+ENG, 2001.

[13] A. H. Abdul Hafez and C. V. Jawahar, “Target model estimation using particle filters for visual servoing,” in *IEEE Int. Conference on Pattern Recognition, ICPR’06*, Hong Kong, August 2006.

[14] S. Hutchinson, G. Hager, and Cork, “A tutorial on visual servo control,” *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, Oct 1996.