

IMAGE BASED VISUAL SERVOING USING BITANGENT POINTS APPLIED TO PLANAR SHAPE ALIGNMENT

Erol Ozgur and Mustafa Unel
Faculty of Engineering and Natural Sciences
Sabanci University
Orhanli Tuzla 34956 Istanbul, Turkey
email: erol@su.sabanciuniv.edu, munel@sabanciuniv.edu

ABSTRACT

We present visual servoing strategies based on bitangents for aligning planar shapes. In order to acquire bitangents we use convex-hull of a curve. Bitangent points are employed in the construction of a feature vector to be used in visual control. Experimental results obtained on a 7 DOF Mitsubishi PA10 robot, verifies the proposed method.

KEY WORDS

Visual servoing, alignment, convex hull, bitangents

1 Introduction

Curve alignment is a central problem in current research areas and has played a key role in many particular domain of applications such as object recognition [1], [2] and tracking [3]. In the domain of visual servoing, most of the current alignment systems are based on known geometrical shaped objects such as industrial parts or those have good features like corners, straight edges which are feasible to extract and track in real time [4]. The alignment of smooth free-form planar objects in unknown environments presents a challenge in visually guided assembly tasks.

In this paper we propose to use bitangent points in aligning planar curves by employing both calibrated [5] and uncalibrated image based visual servoing [6] schemes. In literature the use of bitangents in recognizing planar objects by affine invariant alignment was first considered in [7] and for servoing purposes bitangent lines (lines joining corresponding features on the superposition of two views of a scene) were utilized to align the orientation between two cameras at different locations in space by [8]. In [9] similar principles were applied on landing and surveillance of aerial vehicles using vanishing points and lines. In order to acquire bitangent points, we used convex-hull of a curve [10]. Bitangent points are then used in the construction of a feature vector.

The remainder of this paper is organized as follows: Section 2 presents bitangents of curves and how to acquire them. Section 3 reviews both model based and model free image based visual servoing, for calibrated and uncalibrated approaches. Section 4 is on experimental results for curve alignment and discussions. Finally, Section 5 concludes the paper with some remarks.

2 Bitangents of Curves

A line that is tangent to a curve at two points is called a bitangent and the points of tangency are called bitangent points. See Fig.1. It is well known [11] that these bitangent

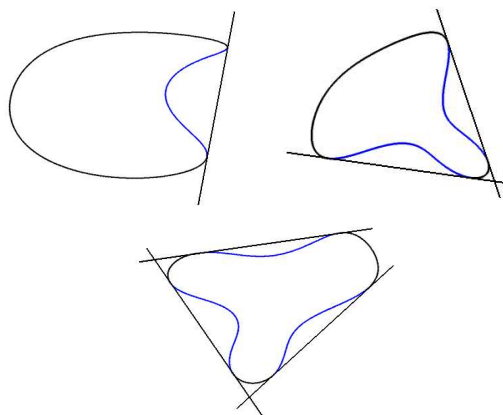


Figure 1. Some curves and their bitangents.

points directly map to one another under projective transformations. They are also called *contact points*.

2.1 Computation of Bitangent Points

Computation of bitangent points of a curve is presented as a block diagram in Fig.2. Block-I receives a sequence of images from a camera and tracks a region in a specified window using a tracking algorithm such as ESM algorithm [12]. Block-II applies Canny edge detection algorithm to the specified region and extracts the curve boundary data. Finally, Block-III employs a Convex-hull algorithm [10] to find convex hull of the curve. Fig.3 depicts a curve with 3 concavities. The convex-hull algorithm yields convex portion of the original data. Initial and final points of each convex portion are bitangent points.

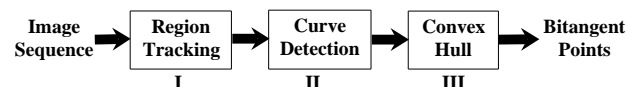


Figure 2. Block diagram representation of the algorithm for extracting bitangent points.

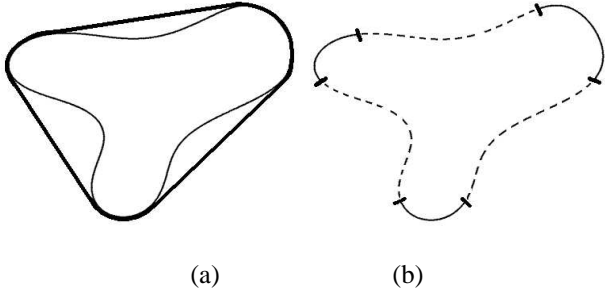


Figure 3. (a) a curve and its convex-hull, (b) convex data portions and the *bitangent* points

3 Visual Servoing

3.1 Background

Let $\theta \in \mathfrak{R}^n$, $s \in \mathfrak{R}^m$ and $r \in \mathfrak{R}^6$ denote the vectors of joint variables of a robot, image features obtained from visual sensors and the pose of the end-effector of the robot, respectively. The differential relation between θ and r with respect to time implies

$$\dot{r} = J_R(\theta)\dot{\theta} \quad (1)$$

where $J_R(\theta) = \partial r / \partial \theta \in \mathfrak{R}^{6 \times n}$ is the robot Jacobian which describes the relation between the robot joint velocities and the velocities of its end-effector in Cartesian space. The relation between s and r is given as $s = s(r)$ and its differentiation with respect to time yields

$$\dot{s} = J_I(r)\dot{r} \quad (2)$$

where $J_I(r) = \partial s / \partial r \in \mathfrak{R}^{m \times 6}$ is the image Jacobian which describes the differential relation of the image features, and pose of the robot end-effector, and the \dot{r} is the camera velocity screw (V_c). The composite Jacobian is defined as

$$J = J_I J_R \quad (3)$$

where $J \in \mathfrak{R}^{m \times n}$ is a matrix which is the product of image and robot Jacobian. Thus, the relation between joint coordinates and image features is given by

$$\dot{s} = J\dot{\theta} \quad (4)$$

3.2 Calibrated Visual Servoing

Let $s^* \in \mathfrak{R}^m$ be the constant desired feature vector and define the error $e \in \mathfrak{R}^m$ on image plane as $e = s - s^*$. Then the control problem can be formulated as follows: design an end-effector velocity screw \mathbf{u} in such a way that the error disappears, i.e. $e \rightarrow 0$.

The image Jacobian of a single point feature vector $s = [x, y]^T$ for a fixed-camera system is given by:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \underbrace{\begin{pmatrix} \frac{1}{Z} & 0 & \frac{-x}{Z} & -xy & (1+x^2) & -y \\ 0 & \frac{1}{Z} & \frac{-y}{Z} & -(1+y^2) & xy & x \end{pmatrix}}_{J_{xy}} V_c \quad (5)$$

where

$$x = \frac{x_p - x_c}{f_x}, \quad y = \frac{y_p - y_c}{f_y} \quad (6)$$

and (x_p, y_p) are pixel coordinates of the image point and (x_c, y_c) are the coordinates of the principle point, and (f_x, f_y) are effective focal lengths of the vision sensor, respectively.

By rearranging and differentiating (6), and writing in matrix form, we get the following expression

$$\begin{pmatrix} \dot{x}_p \\ \dot{y}_p \end{pmatrix} = \begin{pmatrix} f_x & 0 \\ 0 & f_y \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} \quad (7)$$

and substituting (5) into (7), the following equation appears

$$\begin{pmatrix} \dot{x}_p \\ \dot{y}_p \end{pmatrix} = \underbrace{\begin{pmatrix} f_x & 0 \\ 0 & f_y \end{pmatrix}}_{J_I} J_{xy} V_c \quad (8)$$

$$\dot{s} = J_I V_c \quad (9)$$

where J_I is the pixel-image Jacobian. In eye-to-hand case, the image jacobian has to consider the mapping from the camera frame onto the robot control frame. This relationship is given by the robot-to-camera transformation, denoted by:

$$V_c = T V_R \quad (10)$$

where V_R is the end-effector velocity screw in robot control frame. The robot-to-camera velocity transformation $T \in \mathfrak{R}^{6 \times 6}$ is defined as below

$$T = \begin{pmatrix} R & [t]_x R \\ 0_3 & R \end{pmatrix} \quad (11)$$

where $[R, t]$ are being rotational matrix and the translation vector that map camera frame onto robot control frame and $[t]_x$ is the skew symmetric matrix associated with vector t .

Substituting (10) into (9), an expression that relates the image motion to the end-effector velocity is acquired:

$$\dot{s} = \underbrace{J_I T}_{\triangleq \bar{J}_I} V_R = \bar{J}_I V_R \quad (12)$$

where \bar{J}_I is the new image Jacobian which directly relates the changes of the image features to the end-effector velocity in robot control frame. Note that if k feature points are taken into account, e.g. $s = [x_1, y_1 \dots x_k, y_k]^T$, \bar{J}_I is given by the following stacked image Jacobian

$$\bar{J}_I = \begin{pmatrix} \bar{J}_I^1 \\ \vdots \\ \bar{J}_I^k \end{pmatrix} \quad (13)$$

By imposing $\dot{e} = -\Lambda e$ an exponential decrease of the error function is realized. Solving (12), the control for end-effector motion is obtained as follow:

$$V_R = -\bar{J}_I^\dagger \Lambda (s - s^*) \quad (14)$$

where $\Lambda \in \mathfrak{R}^{6 \times 6}$ is a positive constant gain matrix, \hat{J}_l^\dagger is the pseudo-inverse of the image Jacobian and $V_R = (V_x \ V_y \ V_z \ \Omega_x \ \Omega_y \ \Omega_z)^T$.

3.3 Uncalibrated Visual Servoing

Here the composite Jacobian is unknown and it has to be estimated dynamically. The error function in the image plane for a moving target at position $s^*(t)$ and an end-effector at position $s(\theta)$ is given as

$$e(\theta, t) = s(\theta) - s^*(t) \quad (15)$$

where $s^*(t)$ represents desired image features at time t . The control problem can be formulated as follows: design a controller that computes the velocity of joint variables \mathbf{u} in such a way that the error disappears, i.e. $e \rightarrow 0$.

3.3.1 Dynamic Jacobian Estimation

Since the system model is assumed to be unknown, a recursive least-squares (RLS) algorithm [6] is used to estimate the composite Jacobian J . This is accomplished by minimizing the following cost function, which is a weighted sum of the changes in the affine model of error over time,

$$\epsilon_k = \sum_{i=0}^{k-1} \lambda^{k-i-1} \|\Delta m_{ki}\|^2 \quad (16)$$

where

$$\Delta m_{ki} = m_k(\theta_i, t_i) - m_i(\theta_i, t_i) \quad (17)$$

with $m_k(\theta, t)$ being an expansion of $m(\theta, t)$, which is the affine model of the error function $e(\theta, t)$, about the k^{th} data point as follows:

$$m_k(\theta, t) = e(\theta_k, t_k) + \hat{J}_k(\theta - \theta_k) + \frac{\partial e_k}{\partial t}(t - t_k) \quad (18)$$

In light of (18), (17) becomes

$$\Delta m_{ki} = e(\theta_k, t_k) - e(\theta_i, t_i) - \frac{\partial e_k}{\partial t}(t_k - t_i) - \hat{J}_k h_{ki}, \quad (19)$$

where $h_{ki} = \theta_k - \theta_i$, the weighting factor λ satisfies $0 < \lambda < 1$, and the unknown variables are the elements of \hat{J}_k .

Solution of the minimization problem yields the following recursive update rule for the composite Jacobian:

$$\hat{J}_k = \hat{J}_{k-1} + (\Delta e - \hat{J}_{k-1} h_\theta - \frac{\partial e_k}{\partial t} h_t) (\lambda + h_\theta^T P_{k-1} h_\theta)^{-1} h_\theta^T P_{k-1} \quad (20)$$

where

$$P_k = \frac{1}{\lambda} (P_{k-1} - P_{k-1} h_\theta (\lambda + h_\theta^T P_{k-1} h_\theta)^{-1} h_\theta^T P_{k-1}) \quad (21)$$

and $h_\theta = \theta_k - \theta_{k-1}$, $h_t = t_k - t_{k-1}$, $\Delta e = e_k - e_{k-1}$, and $e_k = s_k - s_k^*$, which is the difference between the end-effector position and the target position at k^{th} iteration. The term $\frac{\partial e_k}{\partial t}$ predicts the change in the error function for the next iteration, and in the case of a static camera it can directly be estimated from the target image feature vector with a first-order difference:

$$\frac{\partial e_k}{\partial t} \cong -\frac{s_k^* - s_{k-1}^*}{h_t} \quad (22)$$

The weighting factor is $0 < \lambda \leq 1$ and when close to 1 results in a filter with a longer memory. The Jacobian estimate is used in the visual controllers to determine the joint variables θ_k that track the target.

3.3.2 Dynamic Gauss-Newton Controller

The dynamic Gauss-Newton method [6] minimizes the following time varying objective function

$$E(\theta, t) = \frac{1}{2} e^T(\theta, t) e(\theta, t) \quad (23)$$

By minimizing above objective function it computes the joint variables iteratively as follows:

$$\theta_{k+1} = \theta_k - (\hat{J}_k^T \hat{J}_k)^{-1} \hat{J}_k^T (e_k + \frac{\partial e_k}{\partial t} h_t) \quad (24)$$

Control is defined as

$$u_{k+1} = \dot{\theta}_{k+1} = -K_p \hat{J}_k^\dagger (e_k + \frac{\partial e_k}{\partial t} h_t) \quad (25)$$

where K_p and \hat{J}_k^\dagger are some positive proportional gain and the pseudo-inverse of the estimated Jacobian at k^{th} iteration, respectively.

4 Experiments

In this section, experimental results are presented both for calibrated and uncalibrated visual servoing to demonstrate the validity of the proposed scheme.

Experiments were conducted with a 7 DOF Mitsubishi PA10 robot arm and a Unibrain Fire-i400 digital camera. The camera was mounted on a tripod in eye-to-hand configuration in order to observe the motion of the end-effector. The images were digitized at 320×240 resolution. The system setup is shown in Fig.4. The visual control and image processing modules were implemented in VC++ 6.0 using OpenCV library and run on P4 2.26GHz with 1GB ram personal computer.

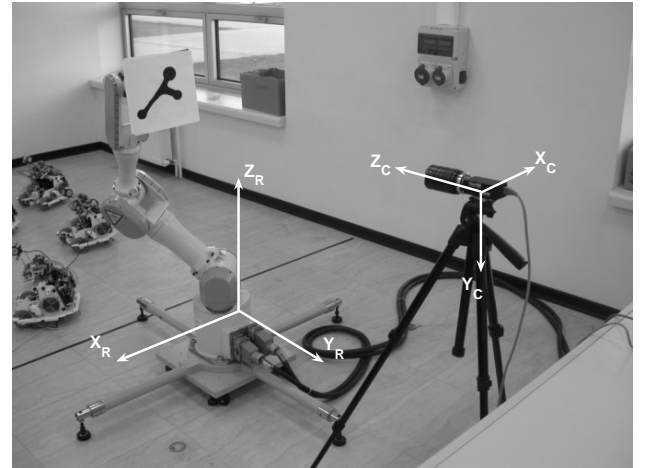


Figure 4. System setup.

Fig. 5 shows a test shape, which is on a plane and rigidly attached to the end-effector. Bitangent points of the shape are acquired using the proposed algorithm in this paper. For visual servoing purposes, either bitangent points or their midpoints, see points denoted by 1, 2 and 3 in Fig. 5, can be used. Unlike bitangent points, which are projective invariant, midpoints are

affine invariant. If the scene's depth is much less than its distance from the camera, a weak-perspective projection can be assumed. Throughout the experiments weak-perspective assumption is made and the visual feature vector s is constructed from the midpoints as follows:

$$s = [x_1, y_1, x_2, y_2, x_3, y_3]^T$$

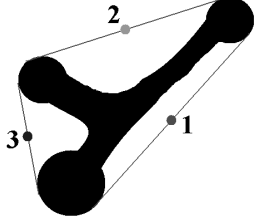


Figure 5. Test shape and midpoints of bitangent points

In the case of perspective projection, i.e. if the weak-perspective assumption does not hold, one can use bitangent points to construct the visual feature vector.

For alignment task, desired pose of the curve is obtained during an off-line stage by moving the robot in xz -plane of the robot control frame with some V_x , V_z and Ω_y for a certain time interval. Consequently, the desired feature vector s^* is constructed from this reference pose.

4.1 Calibrated Visual Servoing Results

The parameters $f_x = 1000$, $f_y = 1000$, $x_c = 160$, $y_c = 120$ are obtained by a coarse calibration of the camera and $Z = 2000$ mm. The robot base frame is positioned at $z = 2000$ mm in z -axis and $y = 1000$ mm in y -axis away from the camera frame. Thus, we have

$$R = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix}, t = \begin{pmatrix} 0 \\ 1000 \\ 2000 \end{pmatrix}$$

where R is the rotational matrix and t is the translational vector that are used for the construction of robot-to-camera transformation matrix T . The gain matrix Λ is tuned as $\Lambda_i = 0.3$ for $i = 1, 2, \dots, 6$. The control input is defined as

$$\mathbf{u} = (V_x \quad V_z \quad \Omega_y)^T$$

where \mathbf{u} consists of the 1st and 3rd components of V_R for the motion in xz -plane and 5th component of V_R for the rotation around y -axis in robot control frame, respectively. Fig.6 depicts the initial and the desired images. Fig.7 shows feature trajectories. Alignment errors and control signals are plotted in Figs.8-9. The norm of the resulting alignment error is found to be less than 1 pixel.

4.2 Uncalibrated Visual Servoing Results

Here we do not need the calibration parameters since the composite Jacobian $J \in \mathbb{R}^{6 \times 3}$ is estimated in a recursive manner. Only 3 joints, namely the 2nd, the 4th and the 6th joints of PA10 robot are used to steer the end-effector by locking the remaining 4 joints. The control parameters are set as $\lambda = 0.96$ and $K_p = 0.6$. The control input is defined as



Figure 6. Initial and desired images

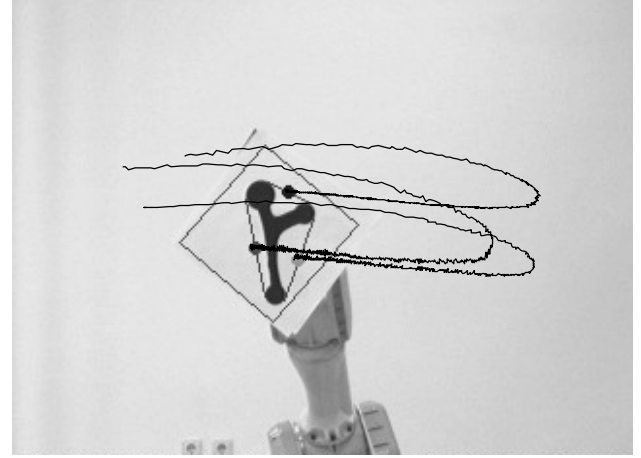


Figure 7. Feature trajectories on the image plane

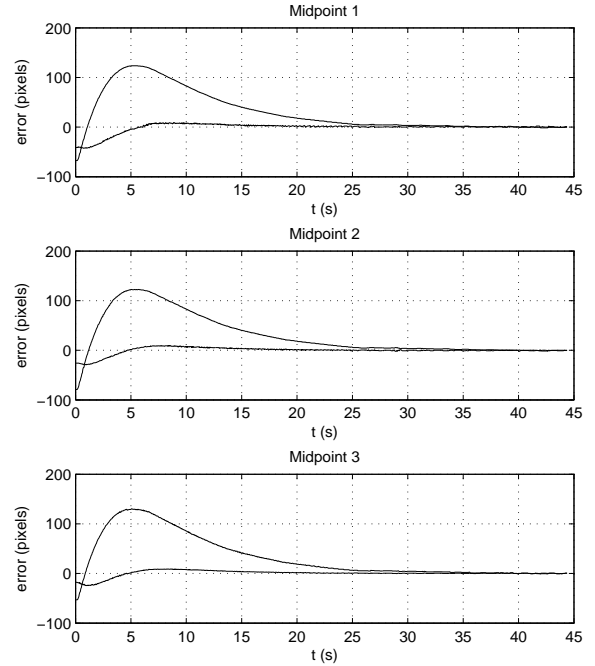


Figure 8. Alignment errors

$$\mathbf{u} = (\Omega_2 \quad \Omega_4 \quad \Omega_6)^T$$

where Ω_2 , Ω_4 and Ω_6 are the joint velocities. Figs.10-11 depict the initial and the desired images, and the feature trajectories on the image plane. Alignment errors and the control signals

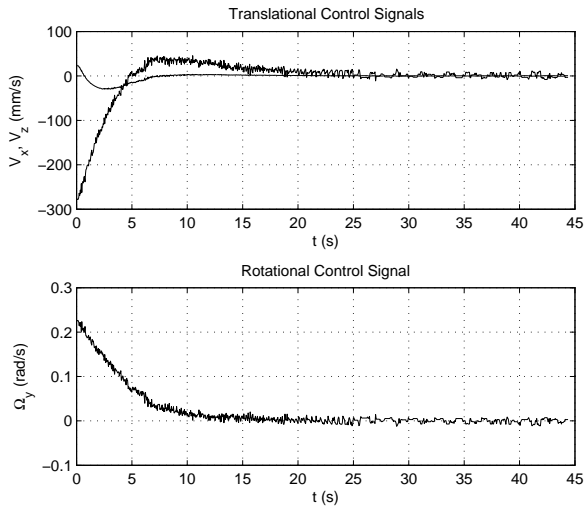


Figure 9. Control signals V_x , V_z and Ω_y

are plotted in Figs.12-13, respectively. The norm of the resulting alignment error is found to be less than 1.5 pixel.

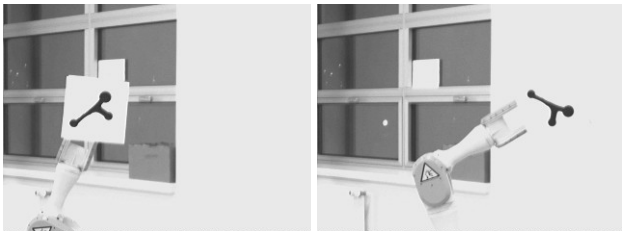


Figure 10. Initial and desired images

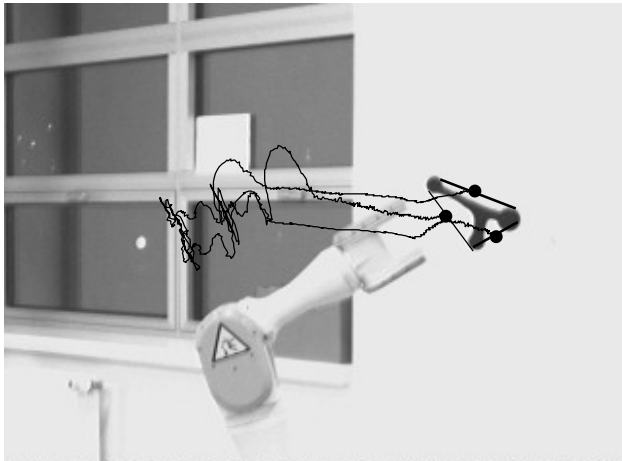


Figure 11. Feature trajectories on the image plane

4.3 Discussions

In both visual servoing approaches we observed that alignment task errors are less than 1.5 pixels, which corresponds to 5 mm in robot workspace. It can be seen that calibrated approach draws

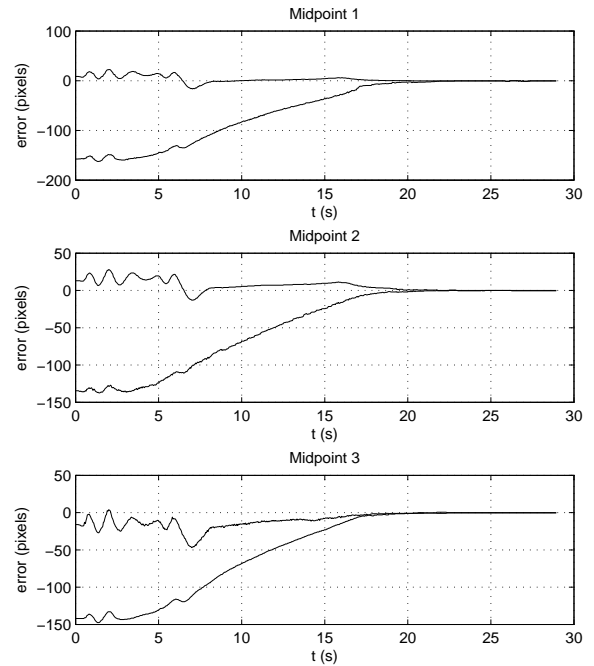


Figure 12. Alignment errors

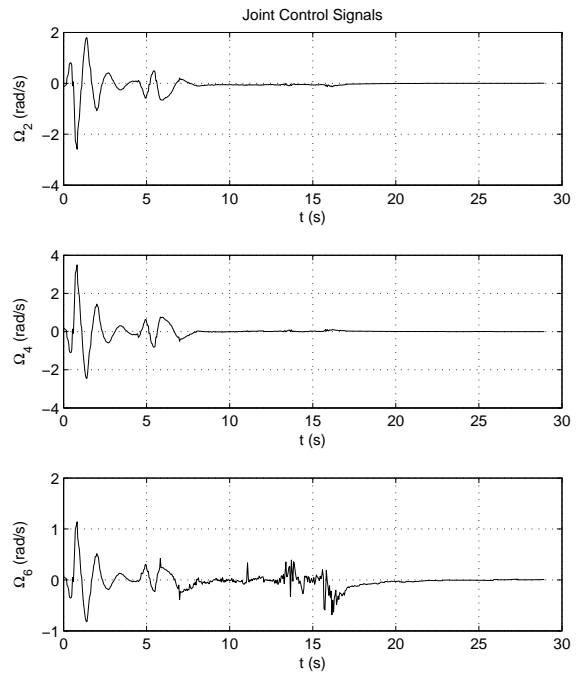


Figure 13. Control signals Ω_2 , Ω_4 and Ω_6

more smoother trajectories while the uncalibrated one shows ambiguous behaviour until the Jacobian converges and the end-effector moves towards the desired pose. Computation times of region tracking, curve detection and bitangent extraction modules are approximately 13 ms, 5 ms and 4 ms, respectively.

5 Conclusion

In this paper, bitangents are used to design image based visual servoing schemes, both calibrated and uncalibrated, for aligning planar shapes with a fixed camera. The assumption is that the curve has at least one concavity on its boundary shape. Experimental results validates the proposed method. Alignment tasks are performed with approximately 5 mm accuracy.

References

- [1] N.J. Ayache and O.D. Faugeras, HYPER: A new approach for the recognition and positioning of two-dimensional objects, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(1), 1986, 4454.
- [2] C. A. Rothwell, A. Zisserman, D. A. Forsyth and J. L. Mundy, Planar Object Recognition using Projective Shape Representation, *International J. of Computer Vision*, 16, 1995, 57-59.
- [3] Hemant D. Tagare, Shape-based non-rigid correspondence with application to heart motion analysis, *IEEE Trans. Medical Imaging*, 18(7), 1999, 570578.
- [4] G. D. Hager. A modular system for robust positioning using feedback from stereo vision. *IEEE Transactions on Robotics and Automation*, 13(4), 1997, 582-595.
- [5] S. Hutchinson, G. D. Hager, and P. I. Corke, A tutorial on visual servo control, *IEEE Trans. on Robotics and Automation*, 12(5), 1996, 651-670.
- [6] J. A. Piepmeier, H. Lipkin, Uncalibrated Eye-in-Hand Visual Servoing, *The International Journal of Robotics Research*, 2003.
- [7] Y.Lamdan, J.T.Schwartz, and H.J. Wolfson. Object recognition by affine invariant matching. *In Proc. CVPR*, pages 1988, 335-344.
- [8] Jacopo Piazzi, Domenico Prattichizzo, Noah J. Cowan, Auto-epipolar Visual Servoing, *International Conference on Intelligent Robots and Systems*, 2004.
- [9] Patrick Rives, Jose R. Azinheira, Linear Structures Following by an Airship using Vanishing Point and Horizon Line in a Visual Servoing Scheme, *International Conference on Robotics & Automation*, 2004.
- [10] J.Sklansky.Measuring concavity on a rectangular mosaic. *IEEE Trans Comput.* 21, 1972, 1355-1364.
- [11] J. L. Mundy, Andrew Zisserman, *Geometric invariance in computer vision*, The MIT Press, 1992.
- [12] S. Benhimane and E. Malis, Real-time image-based tracking of planes using efficient second-order minimization, *IEEE/RSJ International Conference on Intelligent Robots Systems*, Sendai, Japan, October 2004.