Proceeding of the 2013
RSI/ISMInternational Conference on Robotics and Mechatronics
February 13-15, 2013, Tehran, Iran

Visual Tracking in Four Degrees of Freedom using Kernel Projected Measurement

Fateme Bakhshande, *Member, IEEE,* and Hamid D. Taghirad, *Senior Member, IEEE*
Advanced Robotics and Automated Systems (ARAS), Industrial Control Center of Excellence (ICCE),
Faculty of Electrical and Computer Engineering K.N. Toosi University of Technology, Tehran, Iran.
Email: {f.bakhshande, Taghirad}@ieee.org

***Abstract*—Visual Servoing is generally comprised of feature tracking and control. According to the literature, no attempt has already been made to optimize these two parts together. In kernel based visual servoing method, the main objective is to combine and optimize the entire control loop. By kernel definition, a Lyapanov candidate function is formed and the control input is computed so that the Lyapanov stability can be verified. This is performed in four degrees of freedom. In the present study, previous kernel algorithm from the recorded literature has been implemented. We have used the KBVS for our purpose such that an object without any marker is tracked. This method is chosen because of its robustness, speed and featureless properties. Furthermore, in order to show the visual tracking performance, all four degrees of freedom have been synthesized. Experimental results verifies the effectiveness of this method implemented for four degrees of freedom movements.**

I. Introduction

Visual servoing is commonly used for utilizing visual feedbacks to control a robot [1]. Visual servoing (VS) involves moving either a camera or the camera's visual target. The main purpose of this is to track an object in an unknown environment and to converge the target image to a known desired image. In general, visual servoing consists of two parts: feature tracking and control; in addition, these two parts usually work separately in the close-loop system which uses the vision as the underlying parcel of the loop. Therefore, VS is done without tracking and control optimization. When the robot or object moves, features that are extracted from the image are used as the feedback signal, and control sequence is generated based on these features. Therefore the problem can be divided into two sub-problems. By this separation, tuning the whole system together is almost impossible.

In this paper, Kernel Based Visual Servoing (KBVS) method is used to fulfill the VS as a main problem and not divided into two sub-problems. The proposed method has some advantages over previous methods such as position-based and image-based visual servoing [2], [3], 2 1/2 D visual servoing [4] and other advanced methods [5], [6].

KBVS is a new method in which all features of image are used without shrinking the image into limited extracted features. All featureless methods extenuate complex computation because extracting feature points usually requires more computational factors. In addition, these methods attempt to optimize the whole system without separating the tracking and control tasks. Spatial kernel-based tracking algorithm [7], [8], [9], [10] is used in this method for sketching the feedback controller. Lyapunov theory is used to prove the stability of the close-loop system.

The use of kernel method in the VS was introduced in [11], [12], [13]. In these references, the authors defined some kernels based on the spatial weighted average of the image, and then the tracking was shaped by verging on the optimal kernel placement in which the difference in kernel measurement is minimal. Indeed, in this procedure kernel plays the most important role and weighs the feature space of the image. Consequently, it acts as a function that makes samples from the image. By these samples, the control loop is formed and tracking will be completed. Swensen and Kallem rendered different kernels for 2D translation in [12], and also did some experiments on the group of rotations. In [13] Kallem and Dewan proposed a new kernel in depth and roll motions, a Gaussian kernel for depth and a rotationally asymmetric kernel for roll and synthesized them in one kernel. In [11] Swensen and Kallem also analyzed the domain of attraction for some selected kernels through a comparison study. They designated the domain of attraction by acquiring larger area in which the Lyapunov function measurement is positive and its time derivative is negative definite.

In this paper KBVS implementation in four degrees of freedom is presented based on image and Fourier transform traits. Previous work on 2D translation is first explained, the method for translation along and rotation about the z-axis are then elaborated. Furthermore, convergence toward the goal position is analyzed by Lyapunov theory. Experimental results verify suitable performance of the proposed method implemented on four degrees of freedom.

II. Background Material

A. Kernel-based Visual Servoing

In this section, KBVS method is developed. First, tracking in 2D translation parallel to the image plane are demonstrated (x, y). Then, translation along the optical axis (z) and roll about the camera optical axis (θ) have been introduced based on the Fourier transform.

KBVS needs some assumptions to be executable. First, we have assumed that the camera-robot configurations are eye-in-hand configuration that commonly requires fast image processing. Furthermore, a kinematic motion model has been considered for the robot such that joint velocity can be

978-1-4673-5811-8/13/$31.00 2013 IEEE

achieved as the control input. Then, it has been assumed that the target object is planar and camera optical axis is perpendicular to the target plane. Finally, we have assumed that the image scene is continuous and infinite, and also the illumination of the image scene is constant across the image frames.

Consider a signal $S(w,t)$ that is the image intensity for each pixel during the time growth. It is essential that the image at each frame be similar to this signal. Kernel projection value is defined as a function of time, called kernel-projected measurement (KPM) or kernel measurement [12]. It can be expressed as follows:

$$\xi(t) = \int_I K(w)S(w,t)dw. \tag{1}$$

$K \in \mathbf{R}^{n\times 1}$ is the kernel function. Indeed, the image signal $S(w,t)$ is associated with the 2D translation, scale or rotation in the image which are time variants. $w \in \mathbf{R}^2$ is the spatial index parameter for each image captured by the camera. When the camera moves, the amount of the kernel measurement is changed because $S(.,t)$ varies. Assume that S_0 is the goal value for S and ξ_0 is the goal value for ξ. The objective of KBVS is to drive the robot to the goal position or to force $\xi \Rightarrow \xi_0$. In this paper, previous work on 2D translation is first explained, on the basis of which the proposed method for other degrees of motion is then elaborated.

B. Translation Parallel to the Image Plane

Assume that the robot moves parallel to a plane, and we have only a 2D translation motion. Then, the dynamic model for the robot can be expressed as follows:

$$\dot{\mathbf{q}} = \mathbf{u}. \tag{2}$$

Where $\mathbf{q} = [x,y]^T \in \mathbf{R}^2$ is the position of the end effector and $\mathbf{u} \in \mathbf{R}^2$ is the robot control input for each degree of freedom. Note that the control input is the velocity of end effector. As mentioned above, the purpose of KBVS is to drive ξ toward ξ_0. Without loss of generality, let us assume that at ξ_0 the position of the end effector is $x = 0$, $y = 0$ and this point is the goal position. Therefore, the fundamental point is to acquire a control law which drives $[x(t), y(t)]$ toward $[0,0]$. Due to the fact that the distance between the image plane and the target scene is unit and the motion is in a way that they are parallel, it is correct to say that $S(w,q(t)) = S_0(w - q(t))$. Change the coordinate variables by $\bar{w} = w - q(t)$ equation(1) can be written as

$$\begin{aligned}\xi(t) &= \int_I K(\mathbf{w})S_0(\mathbf{w} - \mathbf{q}(t))d\mathbf{w} \\ &= \int_I K(\bar{\mathbf{w}} + \mathbf{q}(t))S_0(\bar{\mathbf{w}})d\bar{\mathbf{w}}.\end{aligned} \tag{3}$$

It has been shown that there is no difference between the case where the kernel is immovable and the image moves or the case where the image is fixed and the kernel moves in the reverse direction. It should be noted that kernel is differentiable but the image signal is not differentiable and we need differentiability of ξ to find a suitable control law.

A Lyapunov function is defined based on KPM error to generate a control law that drives the robot toward the goal position. Hence, a Lyapunov function candidate may be defined as follows:

$$\mathbf{V} = \frac{1}{2} \parallel \xi - \xi_0 \parallel^2 . \tag{4}$$

Using the chain rule, the time derivative of the Lyapunov function may be derived by:

$$\begin{aligned}\dot{V} &= (\xi - \xi_0)^T \frac{\partial \xi}{\partial t} \\ &= (\xi - \xi_0)^T \frac{\partial \xi}{\partial q}\dot{q} \\ &= (\xi - \xi_0)^T (\int_I K'(w)S(w)dw)u.\end{aligned} \tag{5}$$

Where:

$$K'(w) = \frac{\partial K(w)}{\partial w} \in \mathbf{R}^{n\times 2} \tag{6}$$

The following control law ensures that the time derivative of Lyapunov function becomes negative definite.

$$u = -(\int_I \triangledown K(w)S(w,q(t))dw)(\xi - \xi_0), \tag{7}$$

In which,

$$\triangledown K = \frac{\partial K(w)}{\partial w} \in \mathbf{R}^{n\times 2}. \tag{8}$$

And furthermore, the time derivative of Lyapunov function is expressed as:

$$\dot{V} = - \parallel (\int_I \triangledown K(w)S(w,q(t))dw)(\xi - \xi_0) \parallel^2 . \tag{9}$$

C. Translation along the optical axis using Fourier Transform

Fourier transform has been used for separation of translation from scale and rotation. Fourier transform is one of the most useful tools in image processing algorithms. By this transform, any variations in the translation changes to conversions in the phase of image Fourier transform. Also, any changes in the scale and rotation are transformed into changes in the magnitude of image Fourier transform.

In translation along the optical axis the dynamic model for the robot can be expressed as follows:

$$\dot{z} = u. \tag{10}$$

Without loss of generality, let us assume that I_0 is the image signal at $z = 0$. Based on this assumption the following equations may be written:

$$I(w,z) = I_0(w/z) \tag{11}$$

$$F(v,z) = z^2 F_0(zv), v \in R^2 \tag{12}$$

KPM is defined by equation (13), and the Lyapunov function candidate can be defined as given in (14). By choosing the control input as (15), time derivative of Lyapunov function becomes negative definite.

$$\xi(t) = \int_I K(v)F_(v,z)dv \tag{13}$$

$$V = \frac{1}{2} \parallel \xi - \xi_0 \parallel^2 \tag{14}$$

$$u = (\int_I v^T \bigtriangledown K(v) F(v,z) dv)(\xi - \xi_0). \tag{15}$$

D. Rotation about the optical axis using Fourier Transform

As mentioned in section (II-C) Fourier transform will distinguish translation in x and y direction from scale and rotation. For rotation about the optical axis the magnitude of Fourier transform is used as the designated signal similar to the scale position. In this case the dynamic model for the robot can be written as follows:

$$\dot{\theta} = u. \tag{16}$$

Assume that I_0 is the signal image at the goal position ($\theta = 0$). Therefore I is a rotate version of I_0 and can be written as:

$$I(w, \theta) = I_0(R_\theta w) \tag{17}$$

In which,

$$R_\theta = \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix} \tag{18}$$

Based on (17) the magnitude of Fourier transform for the rotated image:

$$F(v, \theta) = F_0(R_\theta v), v \in R^2 \tag{19}$$

The Lyapunov function candidate is defined similar to that of translation along the optical axis. By choosing the control input as (20), time derivative of Lyapunov function becomes negative definite.

$$u = -(\int_I v^T J \bigtriangledown K(v) F(v,\theta) dv)(\xi - \xi_0). \tag{20}$$

Where:

$$J = R_(\pi/2) = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{21}$$

In this paper, the proposed method based on Fourier transform is implemented and the results are shown in the experimental results.

E. Asymptotic Stability

Heretofore, we have assumed a Lyapunov function candidate that is positive definite and found a control input that makes the derivative of Lyapunov function negative semi-definite. To prove asymptotic stability, it is necessary to show that the derivative of Lyapunov function is negative definite.
Without loss of generality we assume that $q_0 = 0$ is the goal position. Therefore, our aim is to show that V is positive definite and $\dot{V}$ is negative definite. To achieve that, use Taylor expansion of $\xi(t)$ about ξ_0.

$$\xi = \xi_0 + \frac{\partial \xi}{\partial q} q(t) + O(q^2) \tag{22}$$

$$\xi - \xi_0 = Jq(t) + O(q^2) \tag{23}$$

In equation (23) J is Jacobian matrix that is defined as follows:

$$J = \frac{\partial \xi}{\partial q} = \int_I \bigtriangledown K(w) S(w, q(t)) dw \tag{24}$$

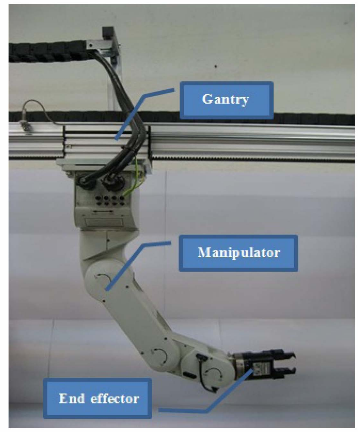

Fig. 1: The robot that we used in our experiments, RV2AJ model

In (22) $O(q^2)$ is high order derivative terms of q. By assuming a small neighborhood around the goal position, high order derivatives can be neglected from the Taylor series.
By using equation (23) Lyapunov function and its derivative are written as follows.

$$\begin{aligned} V &= \frac{1}{2}(\xi - \xi_0)^T P (\xi - \xi_0) \\ &= \frac{1}{2} q(t)^T J^T P J q(t) + O(q^3) \end{aligned} \tag{25}$$

$$Q = J^T P J \tag{26}$$

$$\dot{V} = -q(t) Q Q^T q(t) + O(q^3) \tag{27}$$

In equation (26) P is a $n \times n$ matrix that is positive definite. According to (25) and (27) if Jacobian matrix $J \in \mathbf{R}^{n \times p}$ (in which n is the number of kernels and p is the dimension of q(t)) is a full column rank, then Q is of full rank matrix with $p \times p$ dimension. By this assumption, it can be concluded that V is positive definite and $\dot{V}$ is negative definite in a small neighborhood around the goal position, and furthermore, they are zero at final destination point. Experimental results verifies asymptotic stability behavior near the goal position.

III. Experimental Results

KBVS implementation requires combination of visual tracking in four degrees of freedom. Fourier transform is used for decomposition of 2D translation from scale and rotation. Consequently the magnitude of Fourier transform is used for scale and rotation compensation.Then, 2D motion compensation is performed based on image signal. In this section, some experiments have been conducted to validate the KBVS according to the control laws described in II-B, II-C and II-D. We have done some experiments to show the features of KBVS method. For illustration, some tests in four degrees of freedom and the combination of them have been designed as follows:
1. 2D Translation in X and Y directions.
2. Translation along and rotation about the optical axis by computing Fourier transform.
3. Decomposition of 2D Translation from rotation and scale

corrections using the magnitude of Fourier transform.
4. Combination of 3D translation (x, y, z) plus roll motion about the optical axis.

A. System Setup and Implementation Issues

The hardware setup is composed of a PC equipped with a Pentium IV (1.8 GHz) processor and a 512 MB of Ram. Furthermore, the camera is made by Unibrain Company with 30fps frame rate and a wide lens with 2.1 mm focal length. A 5DOF robot was used to generate the relative cyclic trajectories in the experiments. This robot is a Mitsubishi manipulator model RV-2AJ. Software was implemented in visual studio by using OpenCV library which includes image processing algorithms. The robot configuration is shown in Fig. 1.
All control inputs are velocities at end effector level. By using the robot Jacobian matrix at velocity level all control inputs are transformed to join velocities. For the industrial robots, the control input is not usually executable at the level of joint velocity and the control of the robot is only accessible through the position loop. Furthermore, the robot velocity shall be zero at the start and stop of the each motion. Accordingly, the desired velocity that estimated by kernel method, can not be directly executed. Consequently, the set of desired velocities are transformed to the desired positions by integration at the appropriate time interval. The time interval shall be carefully selected to have a smooth and continuous motion at the outset. Singularity avoidance and joint velocity limitations are considered for implementation based on motion planner that is formerly presented in [14].

B. 2D Translation in X and Y directions

Kernel functions are usually chosen based on the type of experiments [12], [13]. For 2D Translation kernel functions are designed as follows:

$$K_x(v) = \left(\frac{1}{\sqrt{2\pi}\sigma_x}\right)e^{\left(-\frac{(w_1-\mu_x)^2}{2\sigma_x^2}\right)} \tag{28}$$

$$K_y(v) = \left(\frac{1}{\sqrt{2\pi}\sigma_y}\right)e^{\left(-\frac{(w_2-\mu_y)^2}{2\sigma_y^2}\right)} \tag{29}$$

in which, the parameters are set to μ_x=μ_y=-100, σ_x=σ_y=70, while (w_1, w_2) is the image index. In Fig. 2 three random initials for 2D translation are tested. Suitable convergence toward the goal is shown in this figure, while the mean position errors of x and y are mentioned in Table I.

C. Depth and Roll Motion

In this case kernel functions for scale and rotation are selected, respectively, as follows:

$$K_z(v) = e^{-(1/8)\|v\|^2} \tag{30}$$

$$K_\theta(v) = e^{-(1/8)v_1^2} + e^{-(1/8)v_2^2} \tag{31}$$

In Fig. 3 five random initials for scale are tested. Suitable convergence toward the goal is shown in this figure, while the mean position errors of scale test is mentioned in Table I. Fig. 4 shows the five random initials to the goal position for rotation test. Suitable convergence toward the goal is shown in this figure, while the mean position error of θ is mentioned in Table I. As it is observed in these figures, the performance of kernel based visual servoing system is very suitable for different initial conditions. In order to verify similar results a compound motion is considered in the next experiments.

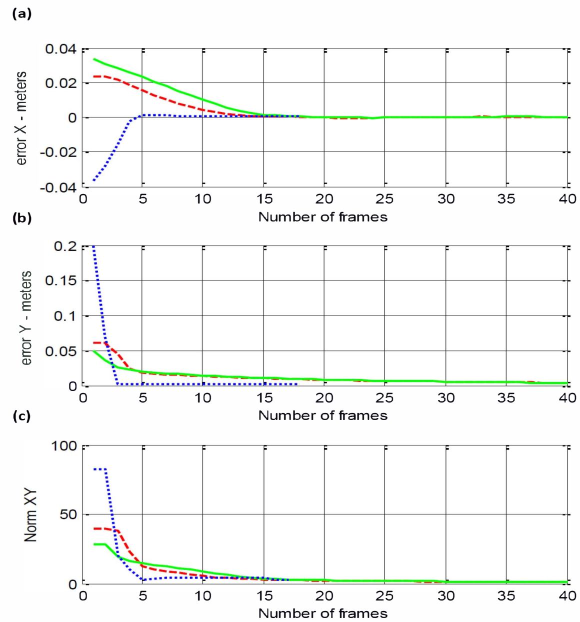

Fig. 2: Three experiment results in x-y directions. a) performance in x axis. b) performance in y axis, c) convergence of KPM for 2d translation

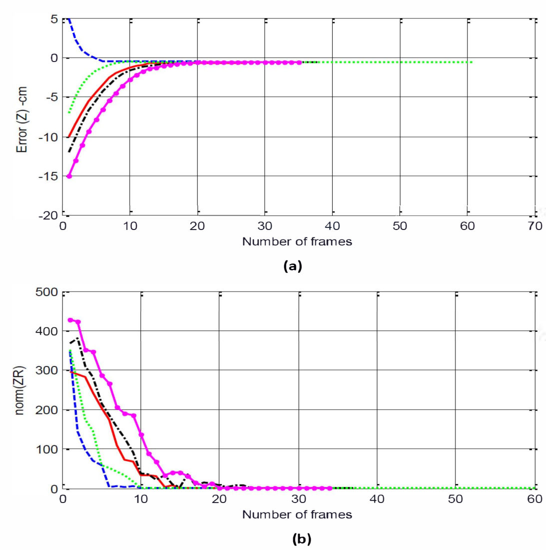

Fig. 3: Five experiment results in z directions. a) performance in z axis. b) Convergence of KPM for z axis using FFT

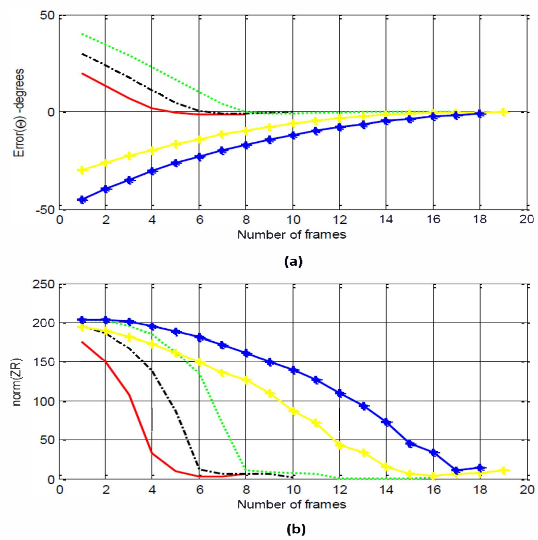

Fig. 4: Five experiment results in rotation about z axis a) performance about θ. b) Convergence of KPM for θ using FFT

D. Decomposition of 2D Translation from rotation and scale corrections

As mentioned for 2D translation the image intensity is used directly but in scale and rotation the magnitude of FFT is required. In this section some experiments are designed to illustrate the effectiveness of the use of FFT in kernel algorithm. Note that in both experiments the magnitude of FFT is independent from 2D translation, and therefore in these experiments, the 2D translation error will not directly compensated.
Fig. 5 illustrates the first experiment where the robot has performed an x - y- z motion. as it is seen in the final picture of this experiment by using FFT of the images in the kernels, the z motion is compensated, but the x and y remains unchanged. Similarly, Fig. 6 illustrates the experiment result for a 3D motion in which in addition to x-y motion rotation along z axis is considered. The same decoupling in motion is clearly observed in the final picture of the target, in which the rotation is compensated for, while the x-y translation is not compensated. Consequently magnitude of FFT is an effective tool to decompose z and θ motions from 2D translation.

E. 3D Translation + Roll Motion

For the final experiment we have considered a full 4D motion, in which the 2D translation in x and y motion is performed in addition to a translation along and a rotation about z axis. In order to perform a full visual servoing motion , first the scale and rotation is compensated by using FFT in the kernels, and then the 2D translation is performed. Fig. 7 illustrates the performance of this experiment, in which the disparity between the final and the goal positions are versy small and hard to be observed in this figure. This result verifies the effectiveness of the decomposition method proposed in this paper based on FFT image inensity. To verify the result quantitatively, Fig. 8 and Fig. 9 are given. Fig. 8 illustrates KPM for 2D translation, rotation and scale, while Fig. 9 demonstrates the position error norms in all four degrees of freedom. As it is shown in these figures, the tracking errors in all 4 degrees of freedom are relatively small, and remain in suitable range. Relative comparison shows similar and better performance in translational motion compared to that of rotational performance.

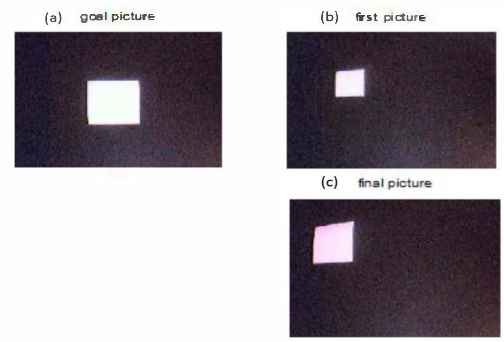

Fig. 5: Example images in a real environment. (a). Goal image. (b). initial image with 2D translation and scale. (c). Final image with scale compensation.

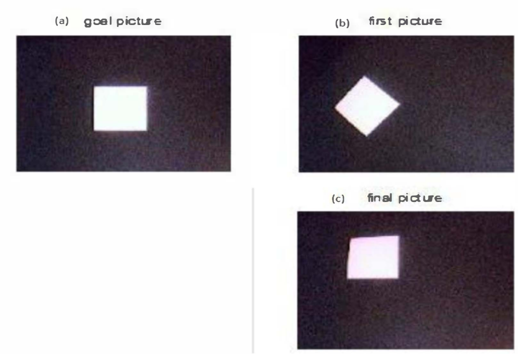

Fig. 6: Example images in a real environment. (a). Goal image. (b). initial image with 2D translation and rotation. (c). Final image with rotation compensation.

TABLE I: Comparison of kernel base tracking by using Fourier transform in scale and rotation

10 trials with random initial positions	position error
mean position error of x (cm)	0.0205
mean position error of y (cm)	0.1202
mean position error of z (cm)	0.5158
mean position error of θ (degrees)	0.2405

IV. Conclusions

Kernel based visual servoing is a method in which tracking is performed based on the KPM as the feedback signal which is a weighted sum of the image. KBVS is a featureless tracking method without the need to separate tracking and control parts.

Fig. 7: Example images of a 4DOF trial in a real environment. The goal, initial, final and disparity image.

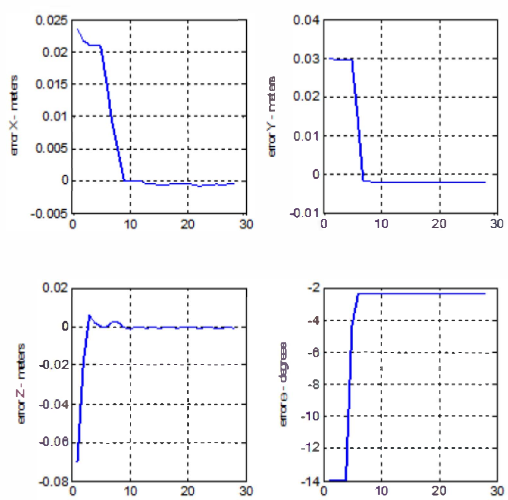

Fig. 8: Trial with random initial position, Convergence in X, Y, Z and R DOF

Based on the KPM, a Lyapanov function is given to verify asymptotic stability of this method. Consequently the convergence of leading an eye-in-hand robot to the goal position without any feature tracking is verified in experiments. In this paper it is proposed to use Fourier transform to decompose 2d translational motion from the motion along, and rotation about the z-axis. Experimental results verifies effectiveness of the proposed method in such decomposition. This idea enables KBVS methods to be concurrently implemented for four degrees of freedom. In the experiments, first the translation along and the rotation about the z axis is compensated by using FFT of image intensity, while at the same time the other 2 degrees of translation are compensated for with the ordinary kernel functions. Final experimental results verifies suitable tracking performance for tracking a unmarked, and non ideal object in a real environment. In the future works comparison between this method and KBVS based on Log-Polar transform

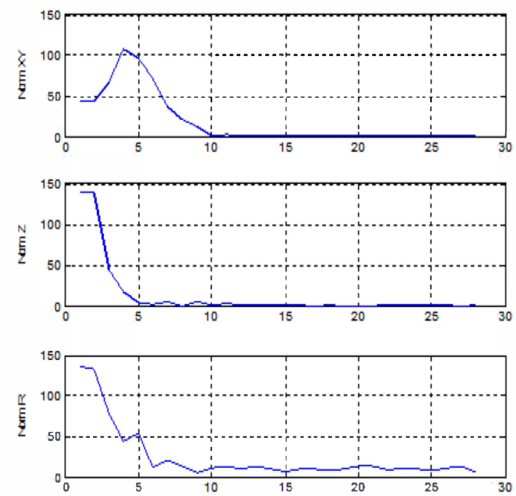

Fig. 9: Trial showing control to the goal image shown in Figure(7). a). Convergence in KPM for X-Y. b). Convergence in KPM for Z. c). Convergence in KPM for R.

would be done to show its sufficiency.

References

[1] A. Castano and S. Hutchinson, "Visual compliance: Task directed visual servo control", IEEE Transactions on Robotics and Automation, 10, pp. 334-342, 1994.

[2] S. Hutchinson, G.D. Hager and P.I. Corke, "A tutorial on visual servo control", 12, pp. 651-670, 1996.

[3] W.J. Wilson, C.C. Williams Hulls and G.S. Bell, "Relative end-effector control using cartesian position based visual servoing", IEEE Transactions on Robotics and Automation, 12, pp. 684-696, 1996

[4] F. Chaumette and E. Malis, "2 1/2 D visual servoing: a possible solution to improve image-based and position-based visual servoings ", IEEE International Conference onRobotics and Automation, 1, pp. 630-635, 2000.

[5] F. Chaumette and S. Hutchinson,"Visual servo control, part II: Advanced approaches", IEEE Robotics and Automation Magazine, 14, pp. 109-118, 2007.

[6] D. Kragic and H.I. Christensen, ", Technical report, Computational Vision and Active Perception Laboratory, 2002.

[7] D. Comaniciu, V. Ramesh and P. Meerc, IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(5):564-575, 2003.

[8] M. Dewan and G. Hager, "Towards optimal kernel-based tracking", In Computer Vision and Pattern Recognition, 1, pp. 618-625, June 2006.

[9] Z. Fan, Y. Wu, and M. Yang, "Multiple collaborative kernel tracking. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2005.

[10] G.D. Hager, M. Dewan, and C.V. Stewart, "Multiple kernel tracking with ssd", In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1, pp. I-790-797 , 2004.

[11] J. Swensen and V. Kallem, and N. Cowan,"Empirical Characterization of Convergence Properties for Kernel-based Visual Servoing", Visual Servoing via Advanced Numerical Methods, SpringerVerlag, pp. 23-38, 2010.

[12] V. Kallem, J.P. Swensen, M. Dewan, G.D. Hager and N.J. Cowan, "Kernel-Based Visual Servoing: Featureless Control using Spatial Sampling Functions".

[13] V. Kallem, M. Dewan, J.P. Swensen, G.D Hager and N.J. Cowan, "Kernel-based visual servoing", In IEEE/RSJ Int. Conf. on Intelligent Robots and System, IROS07, pp. 1975-1980, 2007.

[14] H. Taghirad, M. Shahbazi, F. Atashzar and S. Rayatdoost, "Singular Free Motion Planning in Visual Servoing of Redundant Manipulators", Submitted to IET Computer Vision.