Proceedings of the 2000 IEEE
International Conference on Robotics & Automation
San Francisco, CA • April 2000

Tracking Techniques for Visual Servoing Tasks

D. Kragić[1] and H. I. Christensen[2]

[1]Computer Vision and Active Perception Lab
Dept of Numerical Analysis and Computing Science
Royal Institute of Technology
SE–100 44 Stockholm, Sweden

[2]Centre for Autonomous Systems
Dept of Numerical Analysis and Computing Science
Royal Institute of Technology
SE–100 44 Stockholm, Sweden

Abstract

Many of the todays visual servoing systems rely on use of markers on the object to provide features for control. There is thus a need for a visual system that provides control features regardless of the appearance of the object. Region based tracking is a natural approach since it does not require any special type of features.

In this paper we present two different approaches to region based tracking: a) a multi–resolution gradient based approach (using optical flow) and b) a discrete feature based search approach. We present experiments conducted with both techniques for different types of image motions. Finally, the performance, drawbacks and limitations of used techniques are discussed.

1 Introduction

Visual tracking is extremely useful for robotic interaction in a dynamic world and is assumed to be a solved problem. For visual servoing tasks, tracking is an obvious prerequisite. However, most reported work on visual servoing still relies on use of artificial markers that simplifies figure–ground segmentation and asure robustness over time. Horaud, as an example, in [8] uses four round markers on a robotic gripper to align the gripper with the object to be grasped. A good overview of visual servoing and feature selection can be found in [12].

For operation in a dynamic setting, as for example encountered in service robotics, it is unrealistic to use markers and engineer the environment. For that reason, researchers have adopted other approaches such as region or *feature template* based tracking. A feature template is a 2D entity that represents a portion of an image. During the tracking sequence, the object of interest can be represented by one 2D template or within a multi–template framework where the configuration of individual templates is constrained by some model based information. Two different approaches to region based tracking are considered in this paper: optical flow based tracking and correlation based tracking.

Smith et al. [13], developed a system for detection and tracking of independently moving objects against a non-stationary background. Motion was estimated through tracking of image features (corners and edges) and segmentation was based on an affine motion model. The system was tested on video streams taken from a moving platform (a vehicle traveling along the road).

Brandt, Smith and Papanikolopoulos [2] developed a system using the sum of squared differences (SSD) optical flow measurements as input to the visual control loop. Hager [7] developed the XVision system that has been widely used for manipulation tasks [6]. The system gives a possibility for off–line model selection and performs well when there is good agreement between the model and the actual motion. However, for the case of unexpected object motions the result is usually a loss of tracking. Therefore, there is a need for a system that adaptively selects a motion model in response to current image changes. As pointed out in [9], translational (rigid) motion model gives more reliable results than an affine one when the inter–frame camera motion is small. However, affine changes are necessary to compare distant frames to allow determination of dissimilarity.

In this paper, we present the results obtained for correlation and gradient based approaches to tracking. Experimental results are used for evaluation of the technique and a comparison that determine the best operational characterisitics for each of the techniques. In Section 2 we introduce the different motion models and present the two implemented tracking techniques: region and gradient based. In Section 3 we outline the implemented system and give an overview of the region content evaluation. Experimental results are presented in Section 4. A short summary and conclusions are given in Section 5.

0-7803-5886-4/00/$10.00© 2000 IEEE

2 Image Based Motion

In general, motion estimation problem involves SSD minimization where the minimization process is performed by using a discrete search or Gauss–Newton type of minimization. Both methods assume *intensity constancy* between successive image frames:

$$\mathbf{I}(\mathbf{x}, t+1) = \mathbf{I}(\mathbf{x} - \mathbf{v}(\mathbf{x}, \mathbf{p}), t) \quad (1)$$

where $\mathbf{x} = (x, y)$ is spatial image position of a point, $\mathbf{I}$ is the image intensity, $\mathbf{v}(\mathbf{x}, \mathbf{p})$ denotes image velocity at that point and $\mathbf{p}$ is the number of parameters of the velocity model. We can determine the motion parameters by minimizing the residual:

$$\epsilon = \int\int [\mathbf{I}(\mathbf{x}, t+1) - \mathbf{I}(\mathbf{x} - \mathbf{v}(\mathbf{x}, \mathbf{p}), t)]^2 w(\mathbf{x}) d\mathbf{x} \quad (2)$$

where the summation is performed along the feature window (region of interest) and $w(\mathbf{x})$ is a weighting function that is, in the simplest case, $w(\mathbf{x}) = 1$. In our implementation we used a Gaussian–like function to get rid of a window edge effects. Eq. 2 is a basic structure for computing image motion where the function $\mathbf{v}(\mathbf{x})$ denotes the function of motion that can be parameterized in various ways, e.g. for the affine flow this function will depend on 6 different parameters. To minimize the residual 2, we have to differentiate it with respect to the unknown parameters of the motion model $\mathbf{v}(\mathbf{x})$, see [9] for detailed derivation.

2.1 Models for Image Motion

Depending on the expected 3D motion of the object, we can use one of the following motion models:
Translational 2D motion model:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + T \quad (3)$$

Rigid 2D motion model:

$$\begin{pmatrix} x \\ y \end{pmatrix} = R_\theta \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + T, \; R_\theta = \begin{pmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{pmatrix} \quad (4)$$

Affine 2D motion model:

$$\begin{bmatrix} x \\ y \end{bmatrix} = A \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + T, \; A = \begin{bmatrix} s_x & \gamma \\ 0 & s_y \end{bmatrix} R_\theta \quad (5)$$

where s_x, s_y and γ are scale parameters.

2.2 Correlation vs. Gradient

The use of correlation in pattern matching goes back to the early 60'. It has usually been criticized as computationally costly, prone to errors, and unable to provide a general solution for view–point invariant object recognition [4, 3].There are several types of correlation measures, e.g. direct correlation, mean–normalized correlation, variance-normalized correlation and sum of square differences (SSD). A study of comparison on different correlation methods, performed by Burt, Yen and Xu [3], showed that a direct correlation method and SSD can perform nearly as well as the more complicated methods. An abundance of efforts has been devoted to using different optimization techniques for speeding up the correlation [2, 11]. However, most of the techniques assume only translational changes between frames which rearly happens in a highly dynamic environment such as a robotic workspace.

On the other hand, a multiresolution gradient based approach is a model based approach [1, 9, 7] where the model fitted to the data constrains the overall structure of the estimated motion. This technique can be used to find one [14] or multiple moving targets in the image. It can also be applied to region tracking assuming a constant motion over the region.

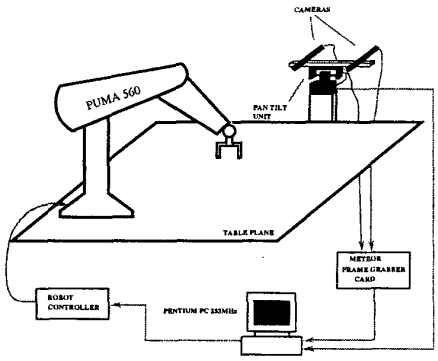


Figure 1: Experimental setup

3 Implementation

Image content or the richness of the region to be tracked is very important for reliable tracking. One of the common problems is the *aperture problem* (in the case of a horizontal edge just a vertical component of motion can be retrieved). For that reason, researchers have proposed different features for tracking (corners, lines, junctions, etc). In general, a good region is the one with a high spatial frequency content.

As proposed in [9] a matrix:

$$\begin{pmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{pmatrix} \tag{6}$$

where g_x and g_y are first derivatives of the image, should be above noise–level and well-conditioned. This implies that the both eigenvalues are large and the order of difference between them should not be large. Applying this to local windows of limited size can result in a large number of candidate regions, especially in the case of salt–and–pepper textures. For that purpose we also added a *cornerity* constraint that gives higher priority to corners and high curvature features. Kitchen and Rosenfeld in [10] define the cornerity κ as follows:

$$\kappa = \frac{g_{xx} g_y^2 + g_{yy} g_x^2 - 2 g_{xy} g_x g_y}{g_x^2 + g_y^2} \tag{7}$$

The features detected this way will be used in the matching process between distant frames to retrieve the affine transformation for the discrete search approach. The features are matched by fast correlation as proposed in [5].

3.1 Discrete Search Approach

We have implemented a matching–based SSD approach combined with a dynamic pyramiding technique and search optimization techniques as proposed in [2]. Proposed optimization techniques are loop short–circuiting, heuristic best–place search position and spiral search. While the mentioned system used the original image as the input signal, we have also tested the performance of our system by using the gradient and the Laplacian of the original image. Since gradient and Laplacian performed worse in the case of high frequency noise, the results presented here are obtained with the raw image data. However, gradient and Laplacian perform well in the case of low frequency noise like reflectance and that should be kept in mind during the implementation stage.

The size of the search window is in direct relationship with the computational cost so it has to be properly selected. Too small search window will not be able to capture the significant position changes, while too large window will result in low tracking frequency. By using dynamic pyramiding approach we are able to perform reliable tracking during large displacement and enhance the positioning accuracy during small displacement. For more details, see [2].

3.2 Gradient Based Approach

Minimizing the objective function 2 for the affine motion model we obtain the following equation:

$$\begin{bmatrix} \Sigma I_x^2 & \Sigma I_x^2 x & \Sigma I_x^2 y & \Sigma I_x I_y & \Sigma I_x I_y x & \Sigma I_x I_y y \\ & \Sigma I_x^2 x^2 & \Sigma I_x^2 xy & \Sigma I_x I_y x & \Sigma I_x I_y x^2 & \Sigma I_x I_y xy \\ & & \Sigma I_x^2 y^2 & \Sigma I_x I_y y & \Sigma I_x I_y xy & I_x I_y y^2 \\ & & & \Sigma I_y^2 & \Sigma I_y^2 x & \Sigma I_y^2 y \\ & & & & \Sigma I_y^2 x^2 & \Sigma I_y^2 xy \\ & & & & & \Sigma I_y^2 x^2 \end{bmatrix} \mathbf{a} = \mathbf{B} \tag{8}$$

where $\mathbf{a} = [a_1, a_2, a_3, a_4, a_5, a_6]^T$ and

$$\mathbf{B} = -\left[\Sigma I_t I_x \;\; \Sigma I_t I_x x \;\; \Sigma I_t I_x y \;\; \Sigma I_t I_y \;\; \Sigma I_t I_y x \;\; \Sigma I_t I_y y\right]^T \tag{9}$$

and matrix on the LHS in the Eq. 8 is symmetric. It is straightforward to see that we are able to use the same equation in the case of translational and rigid motion by choosing just those raws and columns that correspond to the model parameters.

The main difference between the two approaches is that the discrete search approach can bee seen as the global minimization while the gradient based approach is a local minimization method. Therefore, the latter one can result with local minima. In both cases we use Kalman filter in order to predict the position of the region of interest in the next frame, to enable selective processing.

4 Experimental Evaluation

We wanted to test the accuracy of tracking as well the robustness during significant background changes. Therefore, the following experiments were conducted:

- **Experiment 1.** Tracking a planar patch that undergoes translational and rigid motion as explained in Section 2.1.
- **Experiment 2.** Tracking a region that is comprised of a part of a rigid object and its background.
- **Experiment 3.** Tracking a region that undergoes affine motion.

During the experiments, the size of the image window was chosen to 50×50 pixels. The initial size of the search region in the case of discrete search was 4 pixels, but if the displacements were large compared to the search area, the pyramid level was increased. In the case of small displacements, the pyramid level was decreased, i.e. a motion adaptive scale selection was used.

4.1 Experimental Setup

An external camera system was employed with a stereo pair of color CCD cameras, as shown in Fig. 1. In the experiments presented here, the objects being tracked were mounted on a PUMA560 robotic arm. The movement of the arm was under external control.

4.2 Experimental Results

Experiment 1. A planar patch that moves in a plane parallel to the image plane with constant velocity is tracked. The evaluation used XVision, gradient and search based approaches at two different velocities of the manipulator. The results are presented in Fig. 2 and Fig. 3 .

	13 pxl/s	31 pxl/s
Error	Position	Position
MSE Gradient	14.0	17.6
MSE XVision	9.6	28.2
MSE SSD	17.8	36.6
STD Gradient	5.6	5.2
STD XVision	8.8	13.3
STD SSD	4.7	19.6

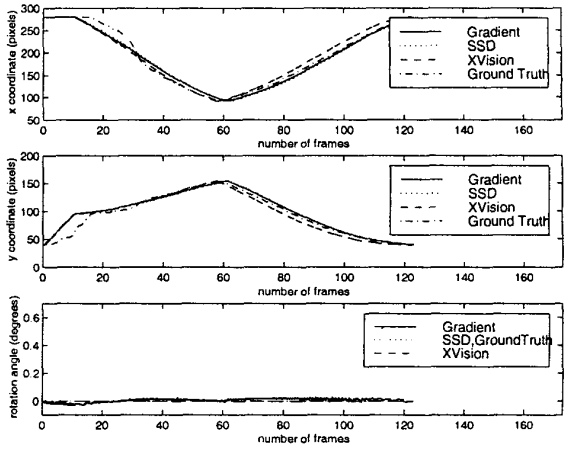
Table 1: Mean squared error and standard deviation for the whole test sequence. The results are presented in Fig. 2

	13 pxl/s	13 pxl/s	20 pxl/s	20 pxl/s
Error	Position	Angle [1]	Position	Angle
MSE Grad.	9.0	-10.8	9.4	-10.1
MSE XV	36.1	-45.4	38.9	-44.6
MSE SSD	9.6	10.2	10.6	9.1
STD Grad.	7.2	28.8	6.0	28.2
STD XV	29.4	13.8	31.0	23.3
STD SSD	7.7	29.3	8.0	27.9

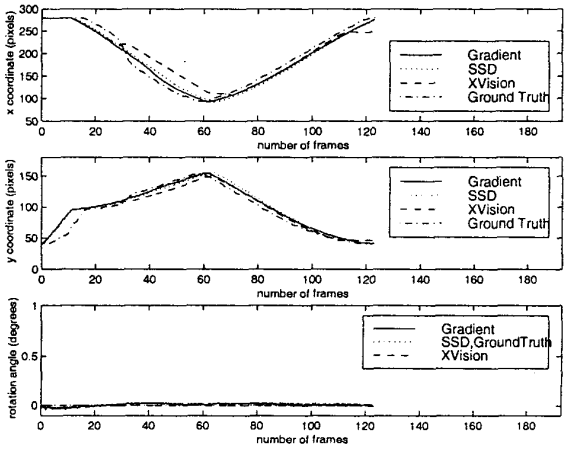
Table 2: Mean squared error and standard deviation for the whole test sequence. The results are presented in Fig. 3.

a) Translation: The manipulator moved so that the motion of the patch was purely translational. The speed in the image coordinates was 13 pixels/s (in average) and 31 pixels/s, respectively. Fig. 2(a) illustrates that all methods maintained tracking of the target object. Changes in motion are modest and this is thus to be expected. For the faster translational motion shown in Fig. 2(b), the XVision system has a significant lag during the frames 30 to 65. As the motion is reversed the lag is gradually eliminated and finally

[1] The angular error is presented as mean error.



(a)



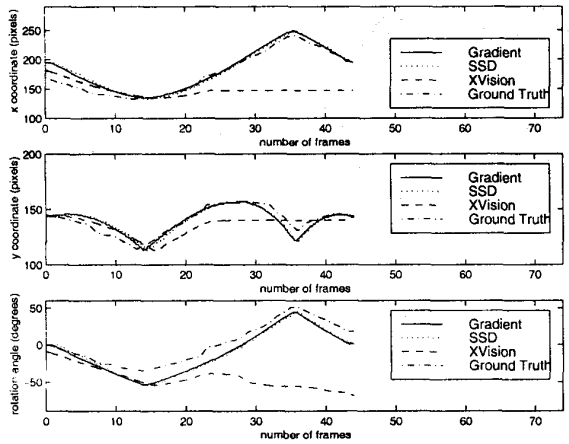
(b)

Figure 2: Translational Motion: the speed of the manipulator was (a)10cm/s (13 pxl/s) and (b)25cm/s (31 pxl/s).

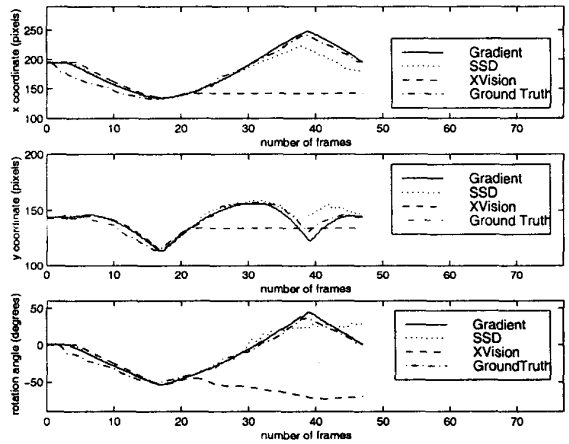
the system converges to the right position when the manipulator brought the patch back into the search window. Mean error and standard deviation in the position are presented in Table 1.

b) Rigid Motion: The manipulator moved so that the patch rotated 50° from the initial position in both directions while translating in the image plane. The speed in image coordinates was 13 pixels/s (in average) and 20 pixels/s, respectively. The results are presented in Fig. 3 and Table 2. After 20th frame, XVision lost the target in both sequences. Fig. 3(b) shows that SSD gave an incorrect response after the 36th frame for the case of increased speed.

The presented results show that both methods per-



(a)



(b)

Figure 3: Rigid Motion: the speed of the manipulator was (a)10cm/s (13 pxl/s) and (b)15cm/s (20 pxl/s).

form well during translational motion. However, the increased size of the tracked region (more than 60×60) adds heavily to the computational complexity for the discrete search approach, which results in loss of tracking or poor accuracy, if we use a high resolution pyramid.

For rigid motion, the increased parameter space for the discrete search approach limits the speed of the algorithm. We are still able to perform the tracking for a low velocity case but the techniques fails when the velocity is increased. We tested the algorithm with a high velocity while decreasing the size of the region and the algorithm performed well. We may conclude that the gradient based approach should be used if the size of the tracked region compared to the image size is large.

Experiment 2. In this experiment, the motion of the patch was translational and the speed of the manipulator was 25cm/s or 31 pixels/s. In addition, the patch comprised a significant amount of background that varied over time, see Fig 4. Fig. 5 and Table 3 illustrate that all methods maintained tracking although XVision deviated from the true position between frames 30 and 50. The discrete search method performed well although the spatial content of the region varied significantly. The reason for this is that during the high velocities (which was the case in this example), the pyramid level was increased and the low resolution image region still contained enough information to maintain the tracking.

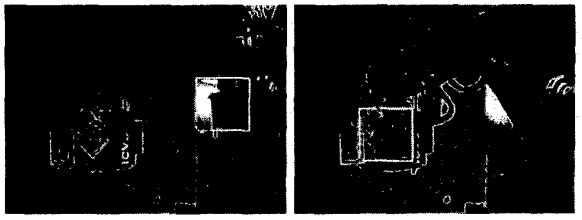


Figure 4: Two example frames during tracking with varying background.

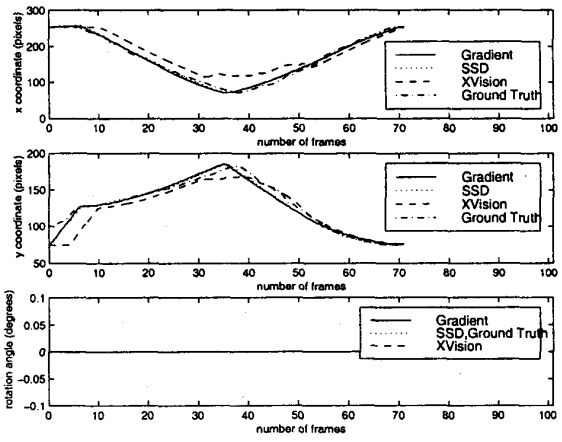


Figure 5: Translational motion with varying background: the speed of manipulator was 25cm/s (31 pxl/s).

Experiment 3. By retrieving affine motion we can detect change in scale and shear in addition to the translational components of the motion and angle.

In the case of discrete search, we retrieve the affine model by matching features between successive frames

Error	Position
MSE Gradient	8.6
MSE XVision	29.2
MSE SSD	6.8
STD Gradient	7.7
STD XVision	7.7
STD SSD	5.3

Table 3: Mean squared error and standard deviation for the whole test sequence. The results are presented in Fig. 5. The velocity in the image plane was 31pxl/s.

as explained in Section 3. In that case, we need a minimum of 3 points but since the data are noisy, it is better to solve an overdetermined system and use (robust) statistics to get rid of the outliers. Some examples can be seen in Fig. 6. In the case of the gradient based approach, rather than solving the 6×6 matrix in Eq. 8, we first solve for translational and rotational motion and, after that, for shear and scale parameters.

Retrieving the six parameters of the affine motion model is usually performed in a Newton–Raphson style minimization procedure[9], which requires a few frames before it converges to the right solution.

In the present implementation, we start with the simplest model and the dissimilarity measure between the first and the current image is used as a measure of goodness of the motion model. We use cross correlation as a dissimilarity measure and maximum thresholds are predefined for each motion model.

The estimated parameters are used to warp the new image. After that, the temporal filtering is performed between the warped and the last image to update the motion parameters. With cross correlation as the dissimilarity measure, we are not able to detect the actual change in the image (lost tracking, occlusion, wrong model) so the future work will be pursued on this issue.

5 Summary and Conclusion

The availability of robust methods for tracking of objects is of tremendous value for robotics in general. It is well known that no single technique is robust to changes in objects appearance and illumination. The methods available have thus been specialized and the environment has been engineered to provide the required robustness. In this paper we have analyzed two different approaches to motion estimation/tracking in combination with three different motion models.
Template based techniques that typically use direct

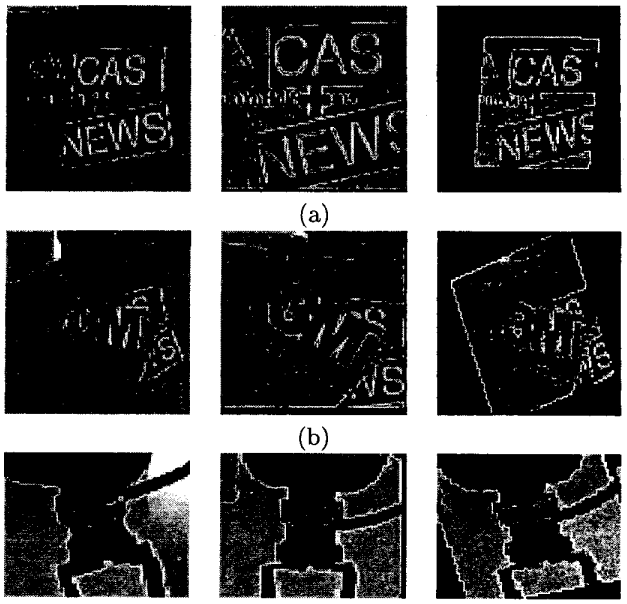


Figure 6: First, last image and the retrieved image for three different motion sequences.

Method	t_x	t_y	θ	s_x	s_y	γ
Gradient	12.2	10.3	0	1.4	1.3	0.006
XVision	11.6	9.9	0	1.4	1.3	0
SSD	13.1	9.8	0	1.4	1.3	0.057

(a)

Method	t_x	t_y	θ	s_x	s_y	γ
Gradient	12	1	0.3	1.3	1.1	0.03
XVision	12.3	0.7	0.3	1.3	1.1	0.02
SSD	15	0.8	0.3	1.3	1.1	0.13

(b)

Method	t_x	t_y	θ	s_x	s_y	γ
Gradient	11	0	0.2	0.9	0.9	0.01
XVision	14.5	1	0.3	0.9	0.8	0.02
SSD	13	1	0.2	0.9	0.9	-0.03

(c)

Table 4: Retrieved affine parameters between the first and the last frame for the three image sequences showed in Fig. 6

image comparison as a basis (with a metric like SSD) are well suited for small image changes as handling of large changes requires search of significant parameters spaces (in particular for general 3D motion). The method is easy to implement. One can expect that the method will perform well for limited size region that has a limited motion. To eliminate the need for au-

tomatic updating of templates it is desirable that the illumination, spatial content, etc. are quasi-static and the object is supposed to be planar or at least convex. An alternative to template based matching is fitting of a motion model to a gradient field (the motion field). Traditionally, motion fields have been very noise sensitive as minimization over small regions results in noisy estimates. For larger regions (assuming rigid motion of a single object) the motion estimate can be computed using least square techniques, which should provide robust results in the presence of limited noise. For noisy images it might be better to use least median square. The above considerations have been evaluated in a real world scenario.

Template based tracking has involved both a new template matching method and the well known XVision system. In addition, an adaptive gradient based method has been tested. The results clearly illustrate that the template based technique is very powerful for small scale motion of limited complexity. For larger motion it is fairly easy to cheat the systems. The gradient based approach provides very robust motion estimates and through use of a scale selection method it can adaptively select the region of interest to accommodate varying motion. The method is however only robust for estimation over large regions as the estimates otherwise become very sensitive to noise.
Motion is almost never time-invariant and in most cases it involves 3D motion (for real scenes). To accommodate this a more advanced motion model has been introduced (affine motion). The new motion model is well suited for true 3D motion, but it overfits data for simple rigid or translational motion. Through adaptive model selection it is however possible to provide robust results over complex motion sequences. Results on tracking of a robot gripper are very encouraging.
Overall a powerful framework for adaptive model selection and *real-time* tracking of objects has been presented. The method enables tracking of relatively fast moving objects in the presence of clutter, etc. Future research will emphasize generalization of the approach to enable tracking of multiple moving objects or tracking of articulated objects.

Acknowledgment: This research has been sponsored by the Swedish Foundation for Strategic Research through the Centre for Autonomous Systems. The funding is gratefully acknowledged.

References

[1] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani. Hierarchical Model-Based Motion Estimation. *Second ECCV, G.Sandini(Ed.), Lecture Notes in Computer Science*, 588:237–252, May 1992.

[2] S.A. Brandt, C.E. Smith, and N.P. Papanikolopoulos. The Minesota Robotic Visual Tracker: A Flexible Testbed for Vision–Guided Robotic Research. *Proc. IEEE International Conference on Humans, Information and Technology,*, 2:1363–1368, 1994.

[3] P.J. Burt, C.Yen, and X. Xu. Local correlation measures for motion analysis – A comparative study. Technical report, Image Processing Laboratory, Electrical, Compute and Systems Engineering Department, February 1982.

[4] J.L. Crowley and J. Martin. Comparision of correlation techniques. *Conference on Intelligent Autonomous Systems, IAS'95*, March 1995.

[5] C.Sun. Multi-Resolution Rectangular Subregioning Stereo Matching Using Fast Correlation and Dynamic Programming Techniques. Technical report, CSIRO Mathematical and Information Sciences, Locked Bag 17, North Ryde, NSW 2113, Australia, August 1998.

[6] G. Hager. A modular system for robust positioning using feedback from stereo vision. *IEEE TRA*, 13(4):582–595, 1998.

[7] G.D. Hager and K. Toyama. The XVision system: A general–purpose substrate for portable real–time vision applications. *Computer Vision and Image Understanding*, 69(1):23–37.

[8] R. Horaud, F. Dornaika, and B. Espiau. Visually guided object grasping. *IEEE TRA*, 14(4):525–532, 1998.

[9] J.Shi and C. Tomasi. Good features to track. *Proc. IEEE CVPR*, June 1994.

[10] L. Kitchen and A. Rosenfeld. Gray–level corner detector. *Pattern Recognition Letters*, 1(2):95–102, 1982.

[11] S. Nassif and D. Capson. Real–time template matching using cooperative windows. *Electrical and Computer Engineering, IEEE Canadian Conference on Engineering Innovation: Voyage of Discovery*, 2:391–394, 1997.

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

[13] S.M. Smith. ASSET-2 - Real-Time Motion Segmentation and Object Tracking. Technical Report TR95SMS2b, Oxford Cente for Functional Magnetic Resonance Imaging of the Brain (FMRIB), Department of Clinical Vision and Image Processing Group, DRA Chertsey, DERA, UK, 1995.

[14] T. Uhlin, P. Nordlund, A. Maki, and J.-O. Eklundh. Towards an active visual observer. Technical report, Dept. of Numarical Analysis and Computing Science, KTH, Stockholm, Sweden, 1995.