Proceedings of the IAPR International Conference on Pattern Recognition. ICPR'98.
Volume 2 - Pages 1251-1254 - Brisbane - Australia - August 1998

Complex object tracking by visual servoing based on 2D image motion

Armel Crétual François Chaumette Patrick Bouthemy
IRISA / INRIA Rennes, Campus de Beaulieu
35042 Rennes cedex, France
E-mail {acretual, chaumett, bouthemy}@irisa.fr

Abstract

Efficient real-time robotic tasks using a monocular vision system were previously developed with simple objects (e.g. white points on a black background), within a visual servoing context. Due to recent developments, it is now possible to design real-time visual tasks exploiting motion information in the image, estimated by robust algorithms. This paper proposes such an approach to track complex objects, such as a pedestrian. It consists in integrating the measured 2D motion of the object to recover its 2D-position in the image. The principle of the tracking task is to control the camera pan and tilt such that the estimated center of the object appears at the center of the image. Real-time experimental results demonstrate the efficiency and the robustness of the method.

1. Introduction

An obvious application of collaboration between robotics and computer vision, in particular with "eye-in-hand" systems, is the pursuit of a moving object. It may indeed be of great interest to perform surveillance tasks. Several papers have dealt with this issue, on both sides of the problem.

First, research dealing with the robotics aspect is generally not interested in vision problems, but in control strategy. As a consequence, there is often a very strong a priori knowledge on the observed object, in order to validate the control law. Most of these works [3, 6, 8, 14] use a quasi-binary image to easily separate the target from the background.

On another hand, several works have emphasized on the visual processing problem of recovering the target center of gravity (c.o.g.) while using well set-on methods to control this estimated position. For example, methods used in [2, 12] only allow to track a small object, or at the best, an object which covers a much smaller part of the image than the background. A 2D affine motion model is computed between two successive images, and the second image is compensated with the opposite motion. Thresholding the difference between this reconstructed image and the original one gives the position of the object. The idea in [10] is quite the same except compensation is based on the measured motion of the camera and thus larger objects can be tracked. In [16], a corner detection algorithm yields the position of a particular point of the object. Finally, in [1, 4, 9, 11] a stereo-vision system is used to build a 3D model of the object motion.

Recently, it was shown that visual servoing based on dynamic measurements [7, 15, 18] can be exploited in real-time applications. The main interest is that estimation of 2D-motion model does not require any visual marks, but only a sufficiently contrasting texture to reliably measure spatio-temporal gradients. Hence, a single moving object could be tracked by regulating its apparent speed to zero. However, application of the vision-based control in this case raises the problem of increasing the derivation order by one compared to using geometric measurements. This may lead to complex and quite unstable control laws. Therefore, we propose to retrieve the target c.o.g. by integrating its speed. Then, we can apply classical control laws, designed for geometric measurements, in order to keep the moving object of interest at the image center to achieve the pursuit task. In Section 2, we briefly recall how we can estimate the motion model and how we can deduce the object c.o.g.. In Section 3, we detail the whole tracking task and report experimental results. In Section 4, we outline the application of our method to pedestrian tracking. Section 5 contains concluding remarks.

2. Object location from 2D-motion

Our aim is to control the robot motion by classical image-based techniques, but without any a priori knowledge on the image content. The solution proposed is to retrieve geometric features by integrating dynamic measurements over time.

Let us denote $s = (x, y)^T$, the 2D projection at time t of a 3D point M, and $\dot{s}$ its apparent speed in the image. s can obviously be recovered knowing the projection position s_0 at time 0 and the evolution of $\dot{s}$ over time, by:

$$\begin{array}{ll} s = s_0 + \displaystyle\int_0^t \dot{s}\, dt & \text{(in continuous form)} \\ s = s_0 + \displaystyle\sum_{i=1}^{k} \dot{s}_i\, \delta t_i & \text{(in discrete form)} \end{array} \quad (1)$$

with $\dot{s}_i$ being the i$^{\text{th}}$ measurement of $\dot{s}$ and δt_i, the time duration between (i-1)$^{\text{th}}$ and i$^{\text{th}}$ measurements.

The motion model used to approximate speed in the image is an affine one with 6 parameters as follows (see [7, 17]):

$$\left\{ \begin{array}{rcl} \dot{x} & = & a_1 + a_2 x + a_3 y \\ \dot{y} & = & a_4 + a_5 x + a_6 y \end{array} \right. \quad (2)$$

$$\text{with} \quad \begin{cases} a_1 = -\frac{T_x}{Z_p} - \Omega_y & a_2 = \gamma_1 \frac{T_x}{Z_p} + \frac{T_z}{Z_p} \\ a_3 = \gamma_2 \frac{T_x}{Z_p} + \Omega_z & a_4 = -\frac{T_y}{Z_p} + \Omega_x \\ a_5 = \gamma_1 \frac{T_y}{Z_p} - \Omega_z & a_6 = \gamma_2 \frac{T_y}{Z_p} + \frac{T_z}{Z_p} \end{cases}$$

where $T = (T_x, T_y, T_z)$ and $\Omega = (\Omega_x, \Omega_y, \Omega_z)$ respectively represent the translational and the rotational components of the kinematic screw associated with the relative rigid motion between the camera frame and the object frame. $Z = Z_p + \gamma_1 X + \gamma_2 Y$ is the equation of the planar approximation of the object surface around the considered point, expressed in the camera frame. Of course, other models (e.g. constant) could be used to estimate the position of the image center. In fact, there is a necessary compromise to find between accuracy of the estimation and computation load.

Motion model estimation algorithm Motion parameters a_i are computed using the multi-resolution robust estimation method (RMR algorithm) described in [13]. A Gaussian image pyramid is constructed at each instant. Let Θ_t be the vector of the six affine motion model parameters at instant t. On the coarsest level, the first estimation of Θ_t consists in minimizing with respect to Θ_t the following criterion:

$$C(\Theta_t) = \sum_p \rho(r(p, \Theta_t))$$

with $r(p, \Theta_t) = \nabla I(p,t).w_{\Theta_t}(p) + I_t(p,t)$, where points p are all the points of the image, I is the intensity function, ∇I and I_t its spatial gradient and temporal derivative, $w_{\Theta_t}(p)$ is the velocity vector at point p provided by Θ_t. ρ is a robust estimator, we take Tukey's biweight function.

Then, we use an incremental strategy. Let Θ_t^k be the estimate of Θ_t at iteration k. We have $\Theta_t^k = \Theta_t + \Delta\Theta_t^k$. Successive refinements $\Delta\Theta_t^k$ are given by:

$$\Delta\Theta_t^k = \arg\min_{\Delta\Theta_t^k} \sum_p \rho(r'(\Delta\Theta_t^k))$$

with $r'(\Delta\Theta_t^k) = \nabla I(p + w_{\widehat{\Theta}_t^k}(p), t+1).w_{\Delta\Theta_t^k}(p) + I(p + w_{\widehat{\Theta}_t^k}(p), t+1) - I(p,t)$. Then, we get $\widehat{\Theta}_t^k = \widehat{\Theta}_t + \widehat{\Delta\Theta}_t^k$ and we iterate. Estimation at a finer resolution level is initialized by the value obtained at the preceding coarser one.

3. Tracking a moving object

The goal of the tracking task is to control the camera pan and tilt such that a detected mobile object remains projected at the center of the image. We are not interested here in problems such as occlusions or multiple moving objects.

3.1. Detection of the mobile object

The detection of the mobile object has to be first performed to obtain its initial projection mask on the image. Since we do not exploit any a priori information on the target, this detection step is achieved using the only property the object undergoes motion. The camera remaining static until the mobile object is detected, the object location is simply determined by difference between two successive images. In practice, because of noise in the images, we use a local spatial average of image intensities. Then, by considering a threshold difference between two successive averaged images, we get a binary image separating moving zones from static ones. The center of gravity of the mask gives the initial position to be regulated to zero.

3.2. Control law

Once the center of gravity (c.o.g.) of the target estimated from (1), we can resort to a standard control law to realize the regulation of this estimated 2D position.

The desired position s^* of $s = (x, y)^T$ being the image center ($s^* = (0,0)^T$), s can be viewed as the vector of error. The visual servoing goal is then to bring and maintain this error to zero by controlling the camera velocity. To design the control law, we use the relation between the temporal variation of s and the camera motion. As this motion is restricted to rotations around the x and y axes, we get from (2):

$$\dot{s} = L \begin{pmatrix} \Omega_{c,x} \\ \Omega_{c,y} \end{pmatrix} + \frac{\partial s}{\partial t} \quad \text{with } L = \begin{bmatrix} xy & (-1-x^2) \\ (1+y^2) & -xy \end{bmatrix}$$

where $\frac{\partial s}{\partial t}$ represents the 2D motion of the target c.o.g. and Ω_c the camera rotation. Specifying an exponential decay of the error with gain λ ($\dot{s} = -\lambda s$), the control law is given by:

$$\begin{pmatrix} \Omega_{c,x} \\ \Omega_{c,y} \end{pmatrix} = -\lambda L^{-1} s - L^{-1} \widehat{\frac{\partial s}{\partial t}} \tag{3}$$

The first term of this control law allows us to reach convergence when the observed object becomes motionless. To remove the tracking errors due to the object own motion, the second term has to be added and can be estimated as explained in [5] by:

$$\widehat{\frac{\partial s}{\partial t}} = \widehat{\dot{s}} - L\widehat{\Omega_c} \tag{4}$$

where $\widehat{\dot{s}}$ is supplied by (2) and the motion parameters provided by the estimation algorithm, and $\widehat{\Omega_c}$ is the measured camera rotational velocity. As described in [5], the estimation $\widehat{\frac{\partial s}{\partial t}}$ of $\frac{\partial s}{\partial t}$ is filtered using a Kalman filter involving a constant acceleration model with correlated noise.

3.3. Results

The tracking task has been tested with a camera mounted on a pan and tilt cell. Images of size 256×256, acquired by a SunVideo board are processed on an UltraSparc station.

To evaluate the accuracy of our control law, we have to compare the true c.o.g. of the object to the estimated one. As we cannot extract in a real-time scheme an exact measurement of the c.o.g. when dealing with a complex object, the control law has been first tested with a simple target from which we can easily extract geometric features. The position used in our control loop of course remains the estimated

one. The considered object in this first experiment has a black surface with four white disks forming a square (see the initial image on Fig 1a). An image processing algorithm running at video rate delivers the position of the c.o.g. for each disk. The computed displacements of this four centers and the measured time interval between two successive images provide the velocity of each of them, and thus parameters a_i using the linear system (2).

The same kind of experiment has been carried out with a textured square from which no geometric features can be easily computed (see the initial image on Fig 1b). To ensure a processing rate as close as possible to the video rate, only a constant motion model is considered in the RMR algorithm. The processing rate reached is about 20 images per second.

The same initial conditions were taken for the two experiments, concerning positions of the target and of the camera. In both cases, the target is translating along a rail alternatively to the right and to the left at constant speed, with a 4 seconds pause between the two motion phases. First (till iteration 800), the object speed was 8 cm/s and then, it was 30 cm/s. The camera is about 1 m away from the object which is in the field of view before it starts moving, but not necessarily at the center. λ was set to 1.5.

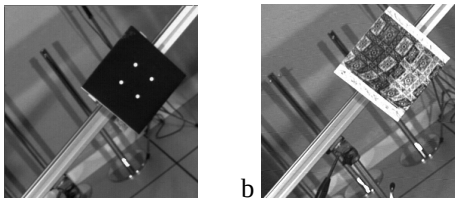

Figure 1. a) Four points object. b) Real square

For the first experiment, the difference between the estimated c.o.g and the true one was always less than 0.5 pixel. Thus, since integrating causes no drift, we can conclude the estimation of the c.o.g. is reliable. Furthermore, previous experiments were conducted to compare estimation errors of the constant parameters between the RMR algorithm and the one used for the four points object. They showed that estimation errors are not greater with the RMR algorithm.

The estimated displacement of the object center for the real square experiment, is plotted in Fig 2. This experiment shows that convergence is correctly obtained for an initial gap of about 40 pixels (it is brought to zero in less than 40 iterations even if the object motion is initially on the opposite direction of the image center). At each abrupt change in the target motion (stop or start), there is an overrun due to the Kalman filter reacting time, but convergence is still obtained.

4. Tracking a pedestrian

The previous application was devoted to the tracking of rigid objects. Let us point out that the estimation of 2D motion parameters with the robust RMR method involves the discarding of non-coherent local motions considered as outliers. Therefore, secondary motions related to deformations of non-rigid objects (such as a human being) do not affect the estimation of the dominant motion. Our pursuit scheme has thus been tested for the tracking of pedestrian.

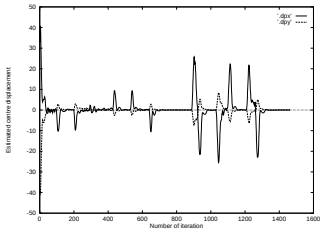

Figure 2. Estimated c.o.g. (in pixel)

Fig 3a displays the mask of a pedestrian resulting from the detection step, where moving zones appear in white. In fact, the estimation uses its dilatation by the structuring element corresponding to a 5×5 pixel square. Fig 3b presents an image acquired during the tracking phase. The white rectangle represents the including rectangle of the detected mask. The motion estimation is done in this window, and the initial weighting values in the IRLS procedure are equal to 1 (resp. 0) if the point belongs (resp. does not belong) to the detected mask. At each iteration, the computation window moves to its predicted position provided by the Kalman filter. The white cross is the estimated c.o.g. of the pedestrian.

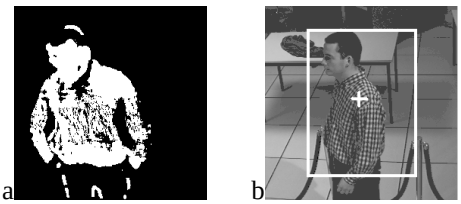

Figure 3. a) Detected mask. b) Pedestrian with the estimation window and estimated c.o.g.

Fig 4 contains one image over 10 (approx. 2 per second) of the sequence processed in real-time by the tracking scheme. Motion of the person is first sideways, and not always facing the camera. Then the pedestrian comes back to the camera. On each image, the estimated c.o.g is represented by a black cross (+) and the image center by a black square ($\diamond$). Despite the complexity of motion and the walk speed (about 4 km/h), the pedestrian always appears at the center. This demonstrates the robustness of the motion estimation algorithm and of the control scheme. Small tracking errors appear, due to always varying 2D speed of the pedestrian and reacting time of the Kalman filter, but are always less than 8 pixels. On images 10 to 13, another person crosses the tracked one. In spite of this perturbing supplementary motion, the camera is still fixating at the selected person.

5. Conclusion

Results presented in this paper have proven that tracking a real object, displaying no prominent features, can be success-

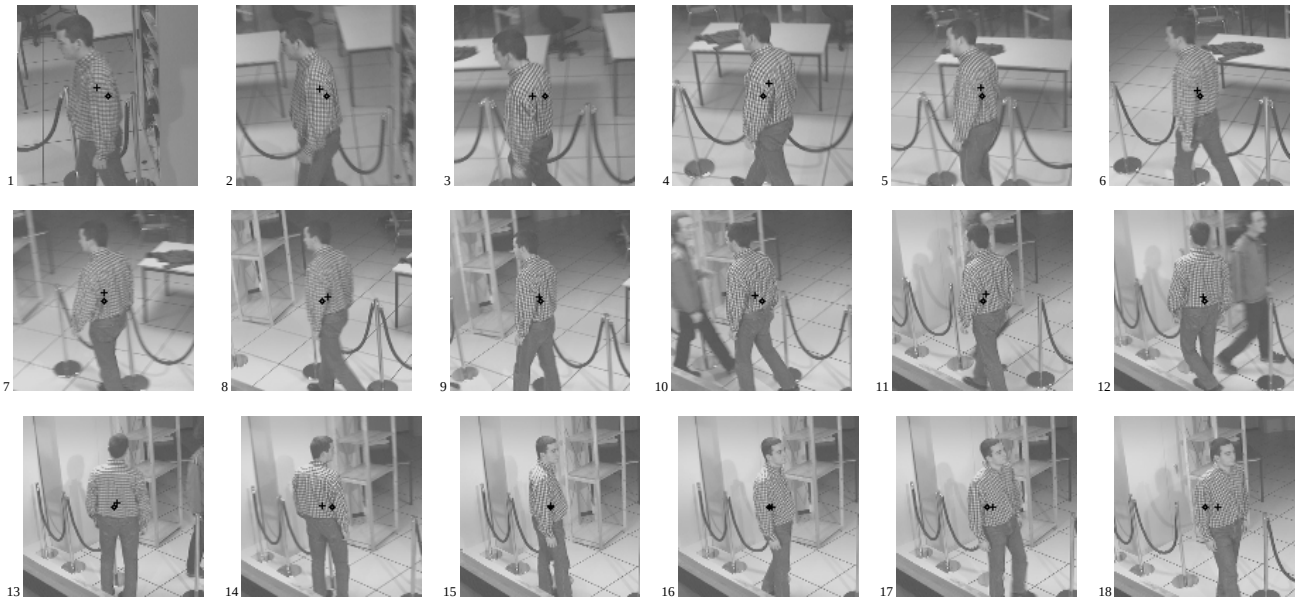

Figure 4. Tracking of a pedestrian. Every 10^{th} frame of the sequence are displayed (approximately 2 frames per second). Symbol + stands for c.o.g of the pedestrian and $\diamond$ for the center of the image

fully achieved by visual servoing. Moreover, this was performed whatever the object size and shape. This is solved by recovering position of the object center by integration of its apparent velocity. The proposed method runs close to video rate. It has been validated for people tracking and provides satisfactory results despite non-rigidity of a human being. If needed this application could be improved with a more sophisticated detection step which could be able to reject masks that do not correspond to a human silhouette.

References

[1] P. Allen, A. Timcenko, B. Yoshimi, and P. Michelman. Automated tracking and grasping of a moving object with a robotic hand-eye system. *IEEE Trans. on Robotics & Automation*, 9(2):152–165, Apr. 1993.

[2] M. Bartholomeus, B. Kröse, and A. Noest. A robust multiresolution vision system for target tracking with a moving camera. In H. Wijshof, editor, *Computer Science in the Netherlands*, pages 52–63. CWI, Amsterdam, Nov. 1993.

[3] F. Bensalah and F. Chaumette. Compensation of abrupt motion changes in target tracking by visual servoing. In *IEEE Int. Conf. on Intelligent Robots & Systems*, volume 1, pages 181–187, Pittsburgh, Aug. 1995.

[4] C. Brown. Gaze control cooperating through prediction. *Image and Vision Computing*, 8(1):10–17, Feb. 1990.

[5] F. Chaumette and A. Santos. Tracking a moving object by visual servoing. In *12^{th} World Congress IFAC*, volume 9, pages 409–414, Sydney, Australia, July 1993.

[6] P. Corke and M. Good. Controller design for high performance visual servoing. In *12th World congress IFAC*, volume 9, pages 395–398, Sydney, Australia, July 1993.

[7] A. Crétual and F. Chaumette. Positionning a camera parallel to a plane using dynamic visual servoing. In *IEEE Int. Conf. on Intelligent Robots & Systems*, volume 1, pages 43–48, Grenoble, France, Sept. 1997.

[8] K. Hashimoto, T. Ebine, K. Sakamoto, and H. Kimura. Full 3D visual tracking with nonlinear model-based control. In *American Control Conference*, pages 3180–3185, San Francisco, California, June 1993.

[9] E. Milios, M. Jenkin, and J. Tsotsos. Design and performance of TRISH, a binocular robot head with torsional eye movements. *Int. Journal of Pattern Recognition & Artificial Intelligence*, 7(1):51–68, Feb. 1993.

[10] D. Murray and A. Basu. Motion tracking with an active camera. *IEEE Trans. on Pattern Analysis & Machine Intelligence*, 16(5):449–459, May 1994.

[11] D. Murray, K. Bradshaw, P. Mc Lauchlan, and P. Sharkey. Driving saccade to pursuit using image motion. *Int. Journal of Computer Vision*, 16(3):205–228, Mar. 1995.

[12] P. Nordlund and T. Uhlin. Closing the loop: detection and pursuit of a moving object by a moving observer. *Image and Vision Computing*, 14(4):265–275, May 1996.

[13] J. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication & Image Representation*, 6(4):348–365, Dec. 1995.

[14] N. Papanikolopoulos, B. Nelson, and P. Khosla. Six d.o.f. hand/eye visual tracking with uncertain parameters. *IEEE Trans. on Robotics & Automation*, 11(5):725–732, Oct. 1995.

[15] P. Questa, E. Grossmann, and G. Sandini. Camera self orientation and docking maneuver using normal flow. In *SPIE AeroSense'95*, Orlando, Florida, Apr. 1995.

[16] I. Reid and D. Murray. Active tracking of foveated feature clusters using affine structure. *Int. Journal of Computer Vision*, 18(1):41, Jan. 1996.

[17] M. Subarrao and A. Waxman. Closed-form solutions to image equations for planar surface in motion. *Computer Vision, Graphics, & Image Processings*, 36(2):208–228, Nov. 1986.

[18] V. Sundareswaran, P. Bouthemy, and F. Chaumette. Exploiting image motion for active vision in a visual servoing framework. *Int. Journal of Robotics Research*, 15(6):629–645, Dec. 1996.