Zoom control to compensate camera translation within a robot egomotion estimation approach

Guillem Alenyà and Carme Torras

Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Llorens i Artigas 4-6, 08028 Barcelona {galenya,torras}@iri.upc.edu

Summary. We previously proposed a method to estimate robot egomotion from the deformation of a contour in the images acquired by a robot-mounted camera [2, 1]. The fact that the contour should always be viewed under weak-perspective conditions limits the applicability of the method. In this paper, we overcome this limitation by controlling the zoom so as to compensate for robot translation along the optic axis. Our control entails minimizing an error signal derived directly from image measurements, without requiring any 3D information. Moreover, contrarily to other 2D control approaches, no point correspondences are needed, since a parametric measure of contour deformation suffices. As a further advantage, the error signal is obtained as a byproduct of egomotion estimation and, therefore, it does not introduce any burden in the computation. Experimental results validate this zooming extension to the method. Moreover, robot translations are correctly computed, including those along the optic axis.

1 Introduction

Zoom control has not received the attention one would expect in view of how it enriches the competences of a vision system. The possibility of changing the size of object projections not only permits analysing objects at a higher resolution, but it also may improve tracking and, therefore, subsequent 3D motion estimation and reconstruction results. Of further interest to us, zoom control enables much larger camera motions, while fixating on the same target, than it would be possible with fixed focal length cameras.

Automating zoom control is, thus, a very promising option for vision systems in general, and robotic applications in particular. One such application, container transfer within a warehouse, where the trajectory of a mobile robot needs to be traced with low precision demands but without any presetting of the environment, has motivated our work [1]. We devised a method to estimate robot egomotion from the image flow captured by an on-board camera. Following the works of Blake [3] and Martínez and Torras [9, 10], instead of using point correspondences, we codify the contour deformation of a selected target in the image with an affine shape vector.

The two main limitations of the method are that, all along the robot trajectory, the target must be kept visible and it should be viewed under weak-perspective conditions (i.e., the depth variation of the target should be small compared to its distance to the camera). Note that, for a robot vehicle such as that of the warehouse, this reduces the set of possible motions almost to just looming and receding. The former limitation can be overcome by mounting the camera on a pan-and-tilt device, while the latter calls for automating zoom control to compensate translation along the optic axis, as addressed in this work.

There are a few papers presenting different strategies for zoom control. Fayman et. al. [4] consider a planar target and robot translations only along the optic axis. In order to keep a constant-sized image projection of the target, they propose a technique, named "zoom tracking", aimed at preserving the ratio f/Z. A thick-lens camera model and full calibration is assumed. Tordoff and Murray [12] address also the problem of fixating the target size in the image, but considering general robot motion, and perspective and affine camera models. With the perspective model, only the case of pure rotating cameras is tackled, as the algorithm needs continuous auto-calibration This algorithm relies also on preserving the ratio f/Z. The authors report some problems for planar targets, far ones, and in situations where perspective effects are not present or discrete, as common in broadcast or surveillance.

We have been investigating the potential of the affine shape representation of the deformation induced by camera motion on an active contour in the image plane [1]. From this representation, egomotion can be recovered, even in the presence of zooming, as will be presented in Sect. 2. In Sect. 3 and 4 our method to recover affine scale and generate zoom demands is introduced. Experimental results are presented in Sect. 5 and finally some conclusions are collected in Sect. 6.

2 Mapping Contour Deformations to Camera Motions

The motion of a robot carrying a camera induces changes in the image due to changes in viewpoint. Under weak-perspective conditions, every

3D motion of the camera results in an affine deformation within the image of the target in the scene. The affinity relating two views is usually computed from a set of point matches [7, 11]. Unfortunately, point matching can be computationally very costly, it being still one of the key bottlenecks in computer vision. Instead, in this work we explore the possibility of using an active contour [3] fitted to a target object. The contour, coded as a B-spline [5], deforms between views leading to changes in the location of the control points.

It has been formerly demonstrated [3, 9, 10] that the difference in terms of control points $\mathbf{Q}' - \mathbf{Q}$ that quantifies the deformation of the contour can be written as a linear combination of six vectors. Using matrix notation

$$\mathbf{Q}' - \mathbf{Q} = \mathbf{WS} \tag{1}$$

where

$$\mathbf{W} = \left(\begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix}, \begin{bmatrix} \mathbf{Q}^\mathbf{x} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{0} \\ \mathbf{Q}^\mathbf{y} \end{bmatrix}, \begin{bmatrix} \mathbf{0} \\ \mathbf{Q}^\mathbf{x} \end{bmatrix}, \begin{bmatrix} \mathbf{Q}^\mathbf{y} \\ \mathbf{0} \end{bmatrix} \right) \tag{2}$$

and $\mathbf{S}$ is a vector with the six coefficients of the linear combination. This so-called shape vector

$$\mathbf{S} = [t_x, t_y, M_{1,1} - 1, M_{2,2} - 1, M_{2,1}, M_{1,2}] \tag{3}$$

encodes the affinity between two views $\mathbf{d}'(s)$ and $\mathbf{d}(s)$ of the planar contour:

$$\mathbf{d}'(s) = \mathbf{Md}(s) + \mathbf{t}, \tag{4}$$

where $\mathbf{M} = [M_{i,j}]$ and $\mathbf{t} = (t_x, t_y)$ are, respectively, the matrix and vector defining the affinity in the plane.

The contour is tracked along the image sequence with a Kalman filter [3] and, for each frame, the shape vector and its associated covariance matrix are updated. Considering a camera that possibly changes the focal length, the affinity coded by the shape vector relates to the 3D camera motion in the following way [10]:

$$\mathbf{M} = \frac{f_i}{f_0} \frac{Z_0}{Z_0 + T_z} \begin{bmatrix} R_{11} & R_{21} \\ R_{21} & R_{22} \end{bmatrix}, \tag{5}$$

$$\mathbf{t} = \frac{f_i}{Z_0 + T_z} \begin{bmatrix} T_x \\ T_y \end{bmatrix} + \begin{bmatrix} u_0 - u_i \\ v_0 - v_i \end{bmatrix}, \tag{6}$$

where R_{ij} are the elements of the 3D rotation matrix $\mathbf{R}$, T_i are the elements of the 3D translation vector $\mathbf{T}$, Z_0 is the distance from the viewed object to the camera in the initial position, f_0 is the focal length at the initial frame, f_i is the current focal length, (u_0, v_0) is the principal point position at the initial frame and (u_1, v_1) is its current position. Using (5) and (6) the deformation of the contour parameterized as a planar affinity permits deriving the camera motion in 3D space. In particular, the scaled translation in direction Z is calculated as [10]

$$\frac{T_z}{Z_0} = \frac{f_i}{f_0} \frac{1}{\sqrt{\lambda_1}} - 1, \tag{7}$$

where λ_1 is the largest eigenvalue of the matrix $\mathbf{MM}^T$. Note that, to simplify the derivation, the reference system has been assumed to be centered on the object.

3 Generating Zoom Demands

Image-based 2D methods rely solely on image measurements. The effect of zooming by a factor f_i/f_0 is to translate the image point u along a line going from the principal point u_0 to the point $x' = \frac{f_i}{f_0}u + (1 - \frac{f_i}{f_0})u_0$. At practical effects, this can be explained as multiplying the calibration matrix corresponding to the first frame by the factor f_i/f_0. Assuming a unit aspect ratio, the scale s of the affinity that relates two views can be recovered from the affine fundamental matrix F_A [6]. Traditionally, it has been estimated from image point correspondences, as the singular vector $\mathbf{N} = (a, b, c, d)^T$ corresponding to the smallest singular value of a matrix constructed with the normalized point correspondences. At least 4 non-coplanar point correspondences are needed.

Instead, with the affinity representation we have introduced, we can estimate the current scale of the affine deformation in relation to the initial contour as a function of the largest singular value λ_1 in the SVD decomposition of $\mathbf{MM}^T$. We propose to use

$$e = \frac{1}{\sqrt{\lambda_1}} - 1. \tag{8}$$

as error function in the zoom control algorithm. It is not directly the affine scale but it is linearly dependent on the related homothecy. Observe that, in the estimation of robot egomotion[1] this error value is already computed, and so, no overcost is added to our process. But we have now the possibility of taking advantage of a zooming camera.

[1] Compared with [12], just motion is recovered (coded as an affine shape deformation), not structure or 3D reprojection, and no foregroung/background extraction is performed, as comes by the definition of the active contour in the tracking algorithm.

Note also that e corresponds to the scaled translation (7) with the ratio value between focal lengths equal to 1. This is effectively a 2D measure. Changing the focal length values with this error function neither calibration is needed nor estimation of the initial distance Z_0. Furthermore, as noticed by Tordoff and Murray [12], the idea of recovering a property of the overall projection instead of a property of the individual points is an advantadge in noisy conditions. This is just what we reach with the weak-perspective camera model and the introduction of the affine shape space.

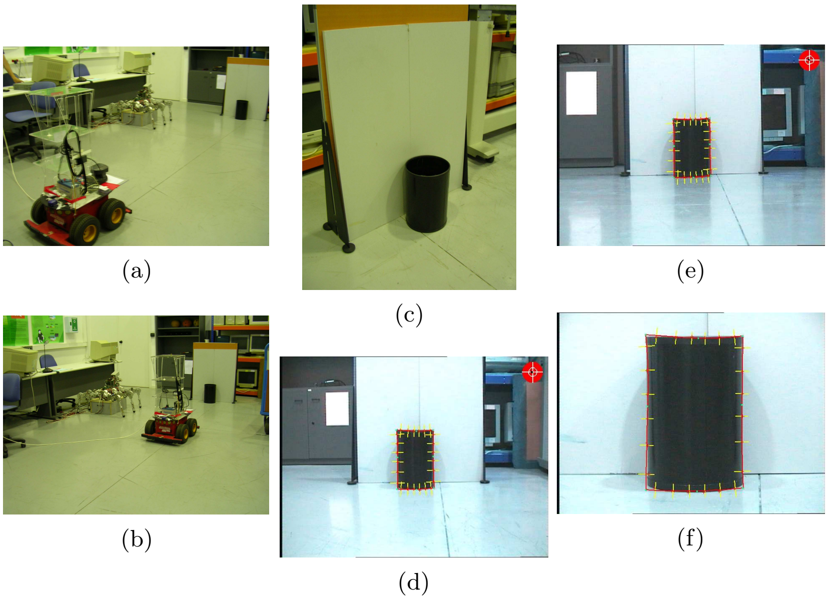
4 Control and Egomotion Algorithm

A zoom control algorithm has been designed to drive the zoom lenses with the proposed error function. In our tests, the velocity-controlled algorithm didn't provide any advantage, as the Visca protocol implemented in the camera only provides a few possible velocities, and this introduces instabilities in the control algorithm. As the precision requirements in terms of translation compensation are not very strict in our system, a simple proportional position controller proved to be enough. We tuned the controller with a very simple process at the beginning of the sequence. After the active contour is initialized, a pulse is induced in the zoom position controller obtaining the relation between the zoom change and the error computed by the error function. Note that no camera calibration and no scene information is needed.

As we are using an uncalibrated camera, the focal length is unknown and so is the ratio of focal lengths f_i/f_0. However, we can use the ratio of zoom positions, as demanded to the zoom mechanism of the camera. We assume that a linear function relates focal length and zoom demand. This is a good approximation when zoom positions are not in the maximum focal length zone [8]. As a consequence, we will not use extreme zoom positions.

When zooming is incorporated to the egomotion algorithm, changes in the sequence of images are obviously produced. As zooming is continuously correcting camera translations, depths in the acquired images are always very similar (this was just our objective!). Compared with a fixed focal length set-up, the difference between the initial contour used as template and the current contour after the deformations induced by camera motion, i.e., the shape vector, are smaller. Ideally, with a very precise and very quick zoom control, the parameters of the affinity codifying the depth translations should be nearly zero. From the zooming factor $\frac{f_i}{f_0}$ introduced, the T_z translation can be recovered.

5 Experimental Results


(a)

(c)

(e)

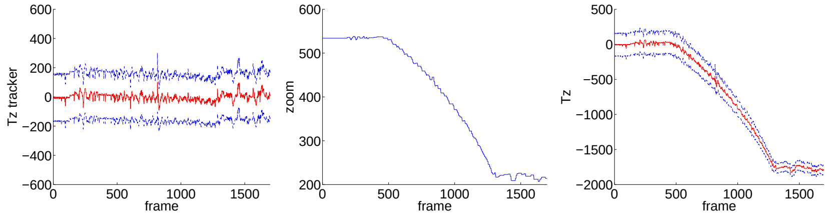
(b)

(d)

(f)

Fig. 1. Images illustrating the performed experiment. **(a)** Robot at the initial position and **(b)** robot at the final position, after performing a translation. **(c)** Detailed image of the target, a common trash can. **(d)** The initial image of the trash can with an active contour fitted to it. **(e)** Image acquired at the final position after the zoom has been changed with the proposed zoom control algorithm. **(f)** Image acquired at the final position without zoom changes

The experiment is performed with a Pioneer AT robot (Fig. 1(a)). It has been equipped with a EVI-D31 pan, tilt and zoom camera. For this experiment, the pan and tilt are kept fixed at a constant value and only zoom is controlled. A linear trajectory is performed with the robot approaching the target. The target used is a common cylindrical trash can (Fig 1(c)). Lines normal to the contour (Fig.1(d)) are search lines along which the tracking algorithm searches peak gradients, with which the shape vector is calculated. While the robot is moving, for each acquired image the tracking algorithm estimates the affine shape deformation of the current contour with respect to the initial one, and computes the egomotion. At frame rate, in our current implementation at 20 fps, the system is capable to generate a zoom demand to cancel in the image the robot translation. Figure 1(e) shows that the zoom



(a) (b) (c)

Fig. 2. Results of the experiment entailing zoom position control. **(a)** Error function in the center, and variance values up and down. **(b)** Zoom position as demanded by the algorithm. **(c)** Reconstructed T_z translation in the center, and variance values up and down

control has effectively cancelled the approaching motion. Figure 1(f) shows the resulting image if the zoom control is deactivated. As can be seen, target projection is much bigger, and after a small approaching translation the target would project out of the image plane.

Figure 2(a) shows the computed error function. Observe that it is centered at 0 and the errors keep always small. In Fig. 2(b) the zoom positions resulting from the zoom control algorithm are plotted. As a trajectory approaching the target is performed, the camera zooms out, leading to lower zoom values. The recovered scaled translation is plotted in Fig. 2(c). Here the initial distance was set to 3500 mm. The recovered translation is scaled, as typical in monocular vision. If we would like to obtain metric information the focal length of the camera should be known. As we are using a zooming camera the relation between the zoom position and the corresponding focal length should be computed.

6 Conclusions and Future Works

Based on the deformation of an active contour fitted to a target, we have shown how to generate zoom control demands that compensate for robot translation along the optic axis, thus keeping the virtual distance from the camera to the target approximately constant. This widens the range of applicability of an algorithm we previously proposed for robot egomotion estimation, in that it permits longer robot translations along the optic axis while preserving weak-perspective conditions.

We use a measure of the scale of the affinity, leading to an algorithm which, as shown in additional experiments not included due to length limitations, is robust to rotations of the robot. No overcost is added to the image manipulation algorithm, as the proposed error measure is already computed as a partial result in order to extract the egomotion.

We are currently working on incorporating pan-and-tilt control to the egomotion recovery algorithm. Preliminary results are promising when we use also the shape vector estimation as error function in the pan-and-tilt control algorithm, as done here for the zoom. This will further lengthen the robot trajectories that our egomotion estimation algorithm could handle.

Acknowledgement. This work is partially funded by the EU PACO-PLUS project FP6-2004-IST-4-27657. The authors thank Gabriel Pi for their contribution in preparing the experiments.

References

1. G. Alenyà, J. Escoda, A.B.Martínez, and C. Torras. Using laser and vision to locate a robot in an industrial environment: A practical experience. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 3539–3544, Apr. 2005.
2. G. Alenyà, E. Martínez, and C. Torras. Fusing visual and inertial sensing to recover robot egomotion. *Journal of Robotic Systems*, 21:23–32, 2004.
3. A. Blake and M. Isard. *Active contours.* Springer, 1998.
4. Jeffrey A. Fayman, Oded Sudarsky, Ehud Rivlin, and Michael Rudzsky. Zoom tracking and its applications. *Machine Vision and Applications*, 13(1):25 – 37, 2001.
5. J. Foley, A. van Dam, S. Feiner, and F. Hughes. *Computer Graphics. Principles and Practice.* Addison-Wesley Publishing Company, 1996.
6. R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2 edition, 2004.
7. J. Koenderink and A. J. van Doorn. Affine structure from motion. *J. Opt. Soc. Am. A*, 8(2):377–385, 1991.
8. M. Li and J.-M. Lavest. Some aspects of zoom-lens camera calibration. *IEEE Trans. Pattern Anal. Machine Intell.*, 18(11):1105–1110, 1996.
9. E. Martínez and C. Torras. Qualitative vision for the guidance of legged robots in unstructured environments. *Pattern Recognition*, 34:1585–1599, 2001.
10. E. Martínez and C. Torras. Contour-based 3d motion recovery while zooming. *Robotics and Autonomous Systems*, 44:219–227, 2003.
11. L. S. Shapiro, A. Zisserman, and M. Brady. 3D motion recovery via affine epipolar geometry. *Int. J. Comput. Vision*, 16(2):147–182, 1995.
12. Ben Tordoff and David Murray. Reactive control of zoom while fixating using perspective and affine cameras. *IEEE Trans. Pattern Anal. Machine Intell.*, 26(1):98–112, January 2004.