Monocular object pose computation with the foveal-peripheral camera of the humanoid robot Armar-III[1]

Guillem ALENYÀ [a] Carme TORRAS [a]
[a] *Institut de robòtica i Informàtica Industrial (CSIC-UPC)*

Abstract. Active contour modelling is useful to fit non-textured objects, and algorithms have been developed to recover the motion of an object and its uncertainty. Here we show that these algorithms can be used also with point features matched in textured objects, and that active contours and point matches complement in a natural way. In the same manner we also show that depth-from-zoom algorithms, developed for zooming cameras, can be exploited also in the foveal-peripheral eye configuration present in the Armar-III humanoid robot.

Keywords. Active contours, motion estimation, depth from zoom

Introduction

Active contours, parameterised as B-splines, are interesting to extract object motion in dynamic scenes because they are easy to track, computation time is reduced when control points are used, and they are robust to partial occlusions. One important property is that a contour is easy to find, even in poorly textured scenes, where most of the point matching algorithms fail. We have been using contours in this way in several works [2,3,4] mainly to recover camera egomotion. One main concern is active contour initialisation. We have always considered that the contour was initialized by an operator. There are several works on contour fitting, but few on automatic contour initialization. Generally, the number of B-splines initialized is too high for general images [10], or either methods require very well segmented images [27].

Here we explore the possibility of using the algorithms developed for active contours with point matches instead. Contrarily to active contours, in order to find enough and reliable point correspondences the objects in which we put attention should be richly textured. As we can see, points and contours complement themselves in a natural way.

It is common to consider Sift features [19] as the state-of-the-art in point detection and matching. Sift features are considered to be invariant to changes in position and orientation, and to scale and illumination up to some degree. Unfortunately, the time required to process a frame, computing all sift features and comparing them with features

[1]This research is partially funded by the EU PACO-PLUS project FP6-2004-IST-4-27657, the Consolider Ingenio 2010 project CSD2007-00018, and the Catalan Research Commission. G. Alenyà was supported by the CSIC under a Jae-Doc Fellowship.

in the model, is too long. Some alternatives have appeared, notably Surf [7] and Fast Keypoints [17]. In this work we aim to assess the feasibility of the developed active contour algorithms when control points are replaced by point matches, and computing time limitations are not taken into account. In the future, to apply this approach to real time scenarios a faster implementations should be developed.

Vision algorithms to recover object or camera motion need the information of the camera internal parameters and initial distance from the camera to the object to obtain metric information. Depth between the camera and the object is usually estimated using two cameras. Unfortunately, maintaining the calibration between both eyes in a lightweight active head, as that present in humanoid robots, is difficult, as head motions and especially saccadic motions produce slight eye motions that are sufficient to uncalibrate the stereo pair. This is the reason why 2D measures are sometimes preferred over 3D reconstruction data when controlling the gaze direction of an active head [26]. We will propose an alternative approach to recover initial depth, which makes use of two cameras with different focal lengths.

This paper is structured as follows. Section 1 presents the principles to compute 3D scene motion from a suitable parameterisation of the motion in the image. In Section 2 the algorithm to compute depth-from-zoom is described. Experiments are presented in Section 3. First, some experiments on depth recovery, and in Section 3.2 a metric reconstruction of an object motion from monocular image streams by making use of the depth recovered in the previous experiment. Finally, conclusions are drawn in Section 4.

1. Pose computation

Object pose involves 3 degrees of freedom (dof) for translation and 3 for orientation. Its computation from point matches is a classical problem in computer vision [12]. A lot of works deal with this problem, in affine viewing conditions [14,18,24] as well as in more general perspective conditions [8,13].

Assuming restricted affine imaging conditions instead of the more general perspective case is advantageous when perspective effects are not present or are minor [25]. The parameterisation of motion as an affine image deformation has been used before for active contour tracking [9], qualitative robot pose estimation [20] and visual servoing [11].

To obtain the pose of an object we use two different views of the same object. We use the first image as the initial position and we relate the motion present in the second image to the first one. If we assume affine imaging conditions, the deformation of the point cloud in the second image with respect to the initial one can be parameterised using a 6 dof *shape vector*, that codifies the possible affine deformations, independently of the number of point matches used. This is an advantadgeous representation because it is more compact and consequently operations to implement tracking algorithms are simpler.

Similarly as has been done in [4], we can consider extracted points $\mathbf{Q}$ as control points of a B-spline, and build a *shape matrix* [20] of the form

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

where $\mathbf{Q^x}$ and $\mathbf{Q^y}$ are the x and y components respectively of the object points in the initial image. When the robot moves, a new image of the object is acquired, and the new point projections $\mathbf{Q}'$ are related to the initial ones $\mathbf{Q}$ through

$$\mathbf{Q}' - \mathbf{Q} = \mathbf{W}\mathbf{S}\,, \tag{2}$$

where

$$\mathbf{S} = (t_x, t_y, M_{11} - 1, M_{22} - 1, M_{21}, M_{12})\ , \tag{3}$$

is the 6-dimensional *shape vector* that encodes the image deformation from the first to the second view. From this shape vector the pose of the camera relative to the initial view can be computed [20] and also its covariance [1]. Algorithm 1 shows how to apply active contour pose estimation using point matches instead of the control points of the B-spline parameterisation of active contours[2].

```
features_initial=Find_object_points(initial_image)
search_tree=Create_search_tree(features_initial)
while Images to treat do
    image_i=capture_image()
    features_frame=Find_points(image_i)
    matches=Search_features(search_tree, features_frame)
    [inliers_model, inliers_frame, num_inliers]=RANSAC(matches)
    if num_inliers>3 then
        [M]=compute_shape_matrix(inliers_initial)
        [S, Σ]=Kalman_filter(inliers_initial, M^-1, inliers_frame)
        pose_i=from_shape_to_3d(S)
        cov_i=propagate_covariance_UT(S, Σ)
    end
end
```

Algorithm 1: Algorithm to use points instead of an active contour in pose computation.

2. Depth estimation from two images taken with different focal lengths

Recently, a depth estimation algorithm has been proposed when a zooming camera is available [4]. Using geometric calibration, the relative depth between the camera and an object can be recovered. With this algorithm there is no need of calibrating the camera intrinsic parameters.

Here we present an alternative approach to that based on a zooming camera. In order to simulate human foveated vision, a common strategy is to use two different cameras [23,22]: one with long focal length simulates the central visual region, with narrow

[2]Compare with algorithms presented in [1].

field of view and high resolution, and another camera with a shorter focal length simulates the peripheral view, with wider field of view and poorer resolution. Other strategies to obtain foveated images include downsampling images [21] and using special lenses [16]. An interesting alternative to camera-based foveal imaging has been proposed recently using a rotating laser scan [15].

Once the robot has oriented the head to see the object with the foveal camera, object projection is into the field of view of both cameras. This situation is equivalent to a zooming camera that takes two images of the same object using two different and known zoom positions.

One assumption was made in the depth-from-zoom algorithm: each zoom controller position should correspond always to the same focal length. In other words, the zoom controller is supposed to have good repetivity. With the presented setup (two cameras per eye), this assumption is no longer necessary, as focal length is fixed for both cameras and never changes.

A common simplification is to consider that the projection centers of both cameras are coincident [26], which is reasonable for some applications. Some zoom camera algorithms also make this assumption, even if it is well known that the optical center changes when zooming. The relation between foveal and peripheral images of a target object is only an affine scaling factor if centers are coincident, and target objects are centered in the projection center of each camera. As we cannot guarantee these facts we cannot restrict deformation only to scalings, and we should allow also for translations in the image plane directions. Thus, we use the reduced shape matrix [4]

$$\mathbf{W_{zoom}} = \left(\begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix}, \begin{bmatrix} \mathbf{Q^x} \\ \mathbf{Q^y} \end{bmatrix} \right) , \tag{4}$$

and compute the reduced shape vector

$$\mathbf{S} = [t_x, t_y, \rho] \ . \tag{5}$$

where ρ is the affine scaling factor.

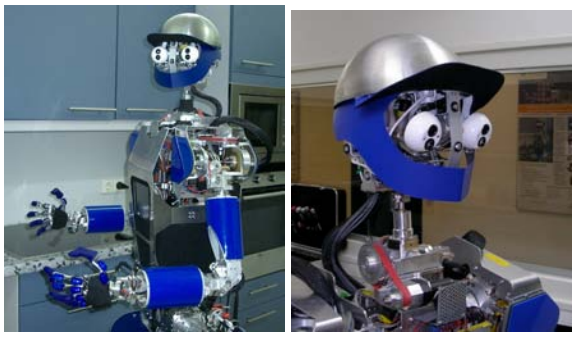
A change in focal length causes a deformation of the object projection in the image that can be parameterised using a shape vector $\mathbf{S}$. If two scaling factors ρ_1 and ρ_2 are computed at two known depths z_1 and z_2 using the same difference in focal lengths, then the unknown depth z of an object can be recovered using the computed scaling factor ρ by applying [1]

$$\frac{z_2 - z_1}{z - z_1} = \frac{\rho(\rho_2 - \rho_1)}{\rho_2(\rho - \rho_1)} . \tag{6}$$

3. Experiments

3.1. Depth estimation in a foveal-peripheral camera configuration

We will use the active head of the Armar-III robot [6] from Karlsruhe University as our experimental platform (Figure 1). The head of Armar-III has two firewire 640×480



(a) Robot torso (b) Head detail, with two cameras *per* eye

Figure 1. Armar III robot used in the experiments

(a) Measured depth=127cm, Actual depth=130cm (b) Measured depth=90.1cm, Actual depth= 90cm

(c) Measured depth=194cm, Actual depth= 200cm (d) Measured depth=159.6cm, Actual depth= 160cm

Figure 2. Peripheral and foveal image pairs (respectively) and the measured depth. Observe that image projection sizes are very similar in each camera for pair 2(a) and 2(b), and also for pair 2(c) and 2(d). This is obviously due to the fact that different object sizes are compensated by the different distances. However, our algorithm successfully recovers actual object depths.

cameras per eye with different focal lengths, each one providing 25fps. The focal length of the foveal camera has been set to acquire focused images from distances larger than 100cm. Observe image defocusing in the foveal image (Figure 2(b)) when the distance from camera to object is only 90cm. We have experimented also with larger distances, but due to defocusing in the foveal images, point position is not precisely measured and, consequently, depth estimation sometimes fails.

Note that for these experiments only one eye, composed of two cameras, has been used. Cameras are not active, so the focal distance is fixed and the depth-of-field also. This is particularly restrictive for the foveal camera, as it has a large focal length. Other methods to obtain depth are difficult to apply here: depth-from-defocus is not applicable as we cannot control the focusing mechanism; depth-from-stereo is badly conditioned, as vergence between peripheral and foveal cameras is nearly 0, and the baseline is very short (aprox. 4mm).

(a) frame0 (b) frame250 (c) frame310 (d) frame450

Figure 3. Peripheral images of the box sequence.

(a) frame0 (b) frame250 (c) frame310 (d) frame450

Figure 4. Foveal images of the box sequence.

Results are depicted in Figure 2, with captions for each experiment. Observe that the sizes in the image of the objects of interest (a cereal box and a small spices jar) in Figures 2(a) and 2(b) are very similar. Actually, they are very different in size but one is closer to the camera that the other. This is the well known depth-size ambiguity in monocular imaging: with no additional information about the object or the scene, all translations can be recovered only up to a scale factor. However, our geometric calibration algorithm successfully recovers the depth, and consequently eliminates this ambiguity[3]. The same applies to Figures 2(c) and 2(d).

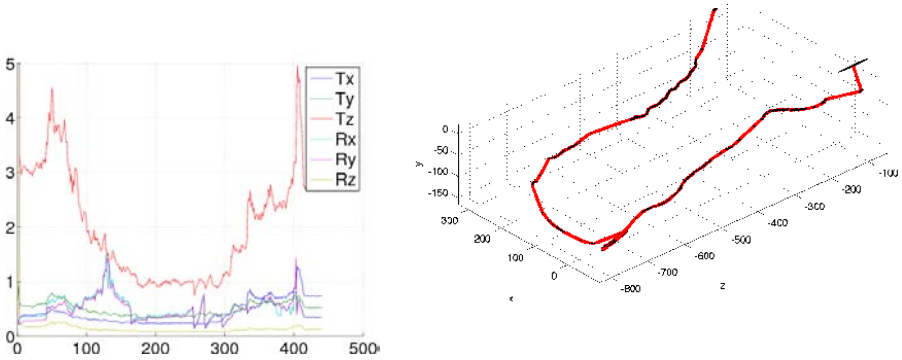
3.2. Metric motion estimation in monocular images

We apply Algorithm 1 presented before to a sequence of a textured object moved freely. Point matches are computed using the Sift algorithm. Figures 3 and 4 show some frames of the video sequence captured by the peripheral and the foveal cameras respectively. As it can be observed, the performed motion includes object translation and rotation.

Motion is estimated using foveal images, as they provide more resolution. Here the cameras are static, but in the future the idea is to control the head motion to maintain the object in the field of view of the foveal camera using the peripheral image.

Results of the motion estimation algorithm can be observed in Figure 5. As expected, variance (Figure 5(a)) is not the same for all pose components. T_z is recovered with higher variance than T_x and T_y, and R_z is recovered with lower variance that R_x and R_x. Observe that when the object approaches the camera, in the middle of the sequence, the variance of T_z diminishes. R_z pose component is always recovered properly despite the change in orientation the object suffered in the middle of the sequence. The path traveled by the object (Figure 5(b)) can be reconstructed metrically using a camera calibration method to recover the focal length of the foveal camera, and with the initial distance recovered with the algorithm presented in Section 2.

[3]Note that in order to obtain metric T_x and T_y translations the focal length should be also known, but it is not necessary for T_z computation.



(a) Variance for each pose component (b) Reconstructed path (red) with covariance (black hyperellipse)

Figure 5. Metric motion estimation for the box sequence.

4. Conclusions

We have shown that it is possible to apply active contour algorithms to recover object motion considering point matches instead of control points in a B-spline parameterisation. Active contours are interesting because they are easy to track and can be used in poorly textured objects. Contrarily, point correspondences are easy to find in richly textured objects. We believe that active contours and point matches complement in a very natural way.

We have presented a depth-from-zooming algorithm that assumes repeatability of the zoom controller: the same zoom controller position corresponds exactly to the same equivalent focal length in the pinhole camera model. Also, the projection rays of the camera when performing the zoom should change in order to observe changes useful for depth estimation. We have shown here that these assumptions can be removed when a foveal-peripheral camera configuration isused, with the advantage that the projection centers are physically not the same.

Acknowledgements

The authors would like to thank Prof. Rüdiger Dillmann and Dr.Tamim Asfour for inviting us to their institute and for the help offered to carry out experiments on the Armar-III robot.

References

[1] G. Alenyà. *Estimació del moviment de robots mitjançant contorns actius*. PhD thesis, Universitat Politècnica de Catalunya, 2007.

[2] 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 *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3539–3544, Barcelona, April 2005.

[3] G. Alenyà, E. Martínez, and C. Torras. Fusing visual and inertial sensing to recover robot egomotion. *Journal of Robotic Systems*, 21(1):23–32, 2004.

[4] G. Alenyà and C. Torras. Depth from the visual motion of a planar target induced by zooming. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4727–4732, 2007.

[5] G. Alenyà and C. Torras. Zoom control to compensate camera translation within a robot egomotion estimation approach. In *Sixth International Workshop on Robot Motion and Control*, volume 360 of *Lecture Notes in Control and Information Sciences*, pages 81–88, 2007.

[6] T. Asfour, K. Regenstein, P. Azad, J. Schröder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann. Armar-III: An integrated humanoid plattfrom for sensory-motor control. In *Proceedings of the 6th IEEE-RAS International Conference on Humanoid Robots*, Genoa, December 2006.

[7] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *Proceedings of the 9th European Conference on Computer Vision*, volume 3951 of *Lecture Notes in Computer Science*, pages 404–417, Graz, 2006. Springer-Verlag.

[8] P.A. Beardsley, A. Zisserman, and D.W. Murray. Sequential updating of projective and affine structure from motion. *International Journal of Computer Vision*, 23(3):235–259, 1997.

[9] A. Blake and M. Isard. *Active contours*. Springer, 1998.

[10] T. Cham and R. Cipolla. Automated b-spline curve representation incorporating mdl and error-minimizing control point insertion strategies. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(1), 1999.

[11] T. Drummond and R. Cipolla. Application of Lie algebras to visual servoing. *International Journal of Computer Vision*, 37(1):21–41, 2000.

[12] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2004.

[13] R. Horaud, F. Dornaika, B. Lamiroy, and S. Christy. Object pose: the link between weak perspective, paraperspective, and full perspective. *International Journal of Computer Vision*, 22(2):173–189, 1997.

[14] J. Koenderink and A. J. van Doorn. Affine structure from motion. *J. Opt. Soc. Am. A*, 8(2):377–385, 1991.

[15] P. Kohlhepp, G. Bretthauer, M. Walther, and R. Dillmann. Using orthogonal surface directions for autonomous 3d-exploration of indoor environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3086–3092, Beijing, October 2006.

[16] K. Kuniyoshi, N. Kita, K. Sugimoto, S. Nakamura, and T. Suehiro. A foveated wide angle lens for active vision. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2982–2988, Nagoya, May 1995.

[17] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1465–1479, 2006.

[18] Y. Liu, T.S. Huang, and O.D. Faugeras. Determination of camera location from 2d to 3d lines and point correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):28–37, 1990.

[19] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[20] E. Martínez and C. Torras. Qualitative vision for the guidance of legged robots in unstructured environments. *Pattern Recognition*, 34(8):1585–1599, 2001.

[21] I.D. Reid and D.W. Murray. Active tracking of foveated feature clusters using affine structure. *International Journal of Computer Vision*, 18(1):41–60, April 1996.

[22] B. Scassellati. A binocular, foveated active vision system. Technical Report 1628, MIT, Artificial Intelligence Laboratory, 1999.

[23] J. Schiehlen and E.D. Dickmanns. Design and control of a camera platform for machine vision. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2058–2063, 1994.

[24] L. S. Shapiro, A. Zisserman, and M. Brady. 3D motion recovery via affine epipolar geometry. *International Journal of Computer Vision*, 16(2):147–182, 1995.

[25] B. Tordoff and D. Murray. Reactive control of zoom while fixating using perspective and affine cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):98–112, January 2004.

[26] A. Ude, C. Gaskett, and G. Cheng. Foveated vision systems with two cameras per eye. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3457– 3462, May 2006.

[27] H. Yang, W. Wang, and J. Sun. Control point adjustment for b-spline curve approximation. *Computer-Aided Design*, 36(7):639–652, 2004.