Project no.:	**IST-FP6-IP-027657**
Project full title:	**Perception, Action & Cognition through Learning of Object-Action Complexes**
Project Acronym:	**PACO-PLUS**
Deliverable no.:	**D7.1.1**
Title of the deliverable:	**Scientific publications on: a) augmented OAC space-time representation, and b) coupled nonlinear estimation and control of OAC algorithms.**

Contractual Date of Delivery to the CEC: **31 July 2007**

Actual Date of Delivery to the CEC: **31 July 2007**

Organisation name of lead contractor for this deliverable: **CSIC**

Author(s): Juan Andrade Cetto, Guillem Alenya, Carme Torras, Michael Villamizar, Alberto Sanfeliu, and Teresa Vidal.

Participants(s): CSIC

Work package contributing to the deliverable: **WP7**

Nature: **R/D**

Version: **1.0**

Total number of pages: **46**

Start date of project: 1^{st} Feb. 2006 **Duration:** 48 month

Projectco-funded by the European Commission within the Sixth Framework Programme (2002-2006)

Dissemination Level

PU	Public	**X**
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Abstract:

This deliverable contains three scientific publications. The first two publications deal with computer vision aspects that arose during the building of an augmented OAC space-time representation. The first publication, entitled “Affine Epipolar Direction from Two Views of a Planar Contour,” is devoted to the geometry of the viewing mechanisms employed. More specifically, it relates changes in object appearance to changes of robot viewpoint, by studying

what can be recovered from two uncalibrated views of a planar contour under affine viewing conditions. The second publication, entitled “Orientation Invariant Features for Multiclass Object Recognition,” discusses one possible set of orientation invariant features that is being explored for the construction of our OAC models.

The third publication, “Active Control for Single Camera SLAM,” treats the problem of moving a camera to incrementally acquire a model of a scene. Active gazing is achieved by maximizing the mutual information between observations and states. Maximizing the mutual information helps the camera avoid ill-conditioned measurements typical of monocular vision systems.

Keyword list: active contours, tracking, orientation invariant features, active control, SLAM.

Table of Contents

1. Introduction

This deliverable focuses on three publications [A, B, and C] relevant to the research that has been performed under WP7.1. The contributions are related to Tasks 7.1.1 and 7.1.2, which deal mainly with action selection for building OAC models. Other published or presented work also relevant to WP7.1 is included in the attached papers [D and E].

2. Camera Motion Estimation

We are interested in building OAC models from varying viewpoints. More than often, accurate camera motion estimates are not available from the kinematics models of the robot moving a camera. This is especially relevant in PACO-PLUS where the ARMAR robot posses redundancy and a complicated kinematics model. Thus, camera motion estimation purely from the content of image views is of major importance. Furthermore, when the objects observed are mostly planar, motion estimation is even more difficult.

2.1 Affine epipolar direction from two views of a planar contour

Most approaches to camera motion estimation from image sequences require matching the projections of at least 4 non-coplanar points in the scene [2]. The case of points lying on a plane has only recently been addressed, using mainly projective cameras. We have studied what can be recovered from two uncalibrated views of a planar contour under affine viewing conditions [6,7]. We proved that the affine epipolar direction can be recovered provided camera motion is free of cyclorotation. The proposed method consists of two steps: 1) computing the affinity between two views by tracking a planar contour, and 2) recovering the epipolar direction by solving a second-order equation on the affinity parameters. Two sets of experiments were performed to evaluate the accuracy of the method. First, synthetic image streams were used to assess the sensitivity of the method to controlled changes in viewing conditions and to image noise. Then, the method was tested under more realistic conditions by using a robot arm to obtain calibrated image streams, which permit comparing our results to ground truth.

3. Feature Extraction

Once the camera motion estimation hurdle has been passed, we concentrated our efforts on the extraction of image features for building OAC models. The objective was to identify a set of features robust to viewpoint change.

3.1 Orientation invariant features for multiclass object recognition

We presented a framework for object recognition based on simple scale and orientation invariant local features that, when combined with a hierarchical multiclass boosting mechanism, produce robust classifiers for a limited number of object classes in cluttered backgrounds [8]. The system extracts the most relevant features from a set of training samples and builds a hierarchical structure of them, by focusing on those features common to all trained objects, and also searching for those features particular to a reduced number of classes, and eventually, to each object class. To allow for efficient rotation invariance, we propose the use of non-Gaussian steerable filters [5], together with an Orientation Integral Image for a speedy computation of local orientation.

4. Action Selection

In building OAC models we must take the actions that are most informative, in the sense that they help reduce the uncertainty in the estimation of OAC feature values. To this end, we devised a strategy to actively choose the most appropriate viewpoint changes when building OAC models using visual primitives.

4.1 Active control for single camera SLAM

Exploring a scene and building a model of the objects contained in it can be interpreted as an instance of the Simultaneous Localization and Mapping problem [4]. SLAM in short is typically addressed as a stochastic state estimation problem, which can be tackled with a variety of filtering techniques. The most popular of them, the Kalman filter [1].

We consider a single hand-held camera performing SLAM at video rate with generic 6DOF motion [3]. The aim is to optimize both the localization of the sensor and the feature map by computing the most appropriate control actions or movements. The actions belong to a discrete set (e.g. go forward, go left, go up, turn right, etc), and are chosen so as to maximize the mutual information gain between posterior states and measurements. Maximizing the mutual information helps the camera avoid ill-conditioned measurements appropriate to bearing-only SLAM. Moreover, orientation changes are determined by maximizing the trace of the Fisher Information Matrix. In this way, we allow the camera to continue looking at those landmarks with large uncertainty, but from better-posed directions. Various position and gaze control strategies are first tested in a simulated environment, and then validated in a video-rate implementation.

5. Attached Papers

[A] Alberich-Carramiñana, M., Alenyà, G., Andrade-Cetto, J., Martínez, E. and Torras, C. (2006) Affine epipolar direction from two views of a planar contour. In *Advanced Concepts for Intelligent Vision Systems*, vol. 4179 of Lecture Notes in Computer Science, pp 944-955, Antwerp, 2006.

[B] Villamizar, M., Sanfeliu, A. and Andrade-Cetto, J. (2006) Orientation invariant features for multiclass object recognition. In *Progress in Pattern Recognition, Image Analysis and Applications*, vol. 4225 of Lecture Notes in Computer Science, pp. 655-664, Cancun, 2006.

[C] Vidal-Calleja, T., Davison, A.D., Andrade-Cetto, J. and Murray. D.W. (2006) Active control for single camera SLAM. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 1930-1936, Orlando, 2006.

[D] Alenyà, G., Alberich-Carramiñana, M. and Torras, C. (2007) Depth from the visual motion of a planar target induced by zooming. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, Rome, 2007. (to appear).

[E] Villamizar, M., Sanfeliu, A. and Andrade-Cetto, J. (2006) Computation of rotation local invariant features using the integral image for real time object detection. In *Proc. IAPR Int. Conf. Pattern Recognition*, vol. 4, pp. 81-85, Hong Kong, 2006.

References

[1] Andrade-Cetto, J., and A. Sanfeliu, The effects of partial observability when building fully correlated maps, IEEE Trans. Robot., vol. 21, no. 4, pp. 771–777, Aug. 2005.

[2] Beardsley, P.A., Zisserman, A., Murray, D.W.: Sequential updating of projective and affine structure from motion. Intl. J. of Computer Vision 23, 1997, 235–259.

[3] Davison, A. J., and D. W. Murray, Simultaneous localisation and mapbuilding uisng active vision, IEEE Trans. Pattern Anal. Machine Intell., vol. 24, no. 7, pp. 865–880, Jul. 2002.

[4] Dissanayake, M. W. M. G., P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, A solution to the simultaneous localization and map building (SLAM) problem, IEEE Trans. Robot. Automat., vol. 17, no. 3, pp. 229–241, Jun. 2001.

[5] Freeman,W.T., Adelson, E.H.: The design and use of steerable filters. IEEE Trans. Pattern Anal. Machine Intell. 13(9) (1991) 891–906.

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

[7] Shapiro, L., Zisserman, A., Brady, M.: 3d motion recovery via affine epipolar geometry. Intl. J. of Computer Vision 16 (1995) 147–182.

[8] Viola, P. and M. Jones, Robust real-time face detection, Int. J. of Computer Vision, vol. 57, n. 2 (2004) 137-154.

Affine Epipolar Direction from Two Views of a Planar Contour*

Maria Alberich-Carramiñana[1], Guillem Alenyà[2], Juan Andrade-Cetto[3],
Elisa Martínez[4], and Carme Torras[2]

[1] Departament de Matemàtica Aplicada I,
UPC Avda. Diagonal 647, 08028 Barcelona
maria.alberich@upc.es
[2] Institut de Robòtica i Informàtica Industrial,
CSIC-UPC Llorens i Artigas 4-6, 08028 Barcelona
{galenya, torras}@iri.upc.edu
[3] Centre de Visió per Computador, UAB Edifici O,
Campus UAB, 08193 Bellaterra, Spain
cetto@cvc.uab.es
[4] CiTS La Salle, Universitat Ramon Llull Pge. Bonanova 8, 08022 Barcelona
elisa@salleurl.edu

Abstract. Most approaches to camera motion estimation from image sequences require matching the projections of at least 4 non-coplanar points in the scene. The case of points lying on a plane has only recently been addressed, using mainly projective cameras. We here study what can be recovered from two uncalibrated views of a *planar contour* under *affine* viewing conditions. We prove that the affine epipolar direction can be recovered provided camera motion is free of cyclorotation. The proposed method consists of two steps: 1) computing the affinity between two views by tracking a planar contour, and 2) recovering the epipolar direction by solving a second-order equation on the affinity parameters. Two sets of experiments were performed to evaluate the accuracy of the method. First, synthetic image streams were used to assess the sensitivity of the method to controlled changes in viewing conditions and to image noise. Then, the method was tested under more realistic conditions by using a robot arm to obtain calibrated image streams, which permit comparing our results to ground truth.

1 Introduction

Recovering camera motion from image streams is an important task in a range of applications including robot navigation and manipulation. This requires a measure of the visual motion on the image plane and a model that relates this motion to the real 3D motion. Most of the existing work on motion recovery relies on a set of point matches to measure visual motion, and, depending on the acquisition conditions, different camera models have been used to emulate the imaging process [1,2]. The full perspective model (the pinhole camera), in

* This work is partially funded by the EU PACO-PLUS project FP6-2004-IST-4-27657.

J. Blanc-Talon et al. (Eds.): ACIVS 2006, LNCS 4179, pp. 944–955, 2006.

either its calibrated (perspective camera) or uncalibrated (projective camera) versions, has proved to be too general when perspective effects diminish. Under weak-perspective viewing conditions (small field of view, or small depth variation in the scene along the line of sight compared to its average distance from the camera), simplified camera models, such as orthographic, scaled-orthographic or their generalization for the uncalibrated case, the affine camera model, provide an advantageous approximation to the pinhole camera, which avoids computing ill-conditioned parameters by explicitly incorporating the ambiguities due to weak perspective into the model.

This paper addresses the motion estimation problem in the context of an affine camera using active contours to measure visual motion. There are several previous motion estimation methods based on affine cameras [3,4]. A common feature of these algorithms is that they require the matching of at least four non-coplanar points and fail for planar structures [5]. The particular case of features lying on planes has not been analyzed in detail thus far. The formulation of this problem is the core of the present paper.

It is well known that two views of a plane are related by a collineation under full perspective projection. Several authors have used this fact to propose algorithms for camera calibration [6], self-calibration [7,8], or extraction of structure and motion from uncalibrated views of points on planes [9] or of planar curves [10]. However, when perspective effects diminish, the relationship between two views of a planar structure becomes an affinity, which invalidates the methods based on collineations.

Following the stratified analysis of motion for affine viewing conditions introduced by Koenderink and van Doorn [3] and revisited by Shapiro et al. [4], we first explore what information of the affine epipolar geometry can be inferred from the affine deformation of the projection of a rigid and planar contour in two weak-perspective views. This sets the basis to derive the motion parameters in a second stage. We show that, under a 3D motion free of cyclorotation, the epipolar direction can be recovered by relating the two affine views of the contour. A series of experiments is performed to test the sensitivity of the method to the different conditions imposed.

The paper is organized as follows. Section 2 contains the analytic study of two weak-perspective views and provides the basis for the recovery of the epipolar direction. Section 3 explains how the parameters of the affinity relating the two views are extracted in our implementation, based on a contour tracker. Section 4 is devoted to experimentation, using both synthetic and real image streams. Finally, Section 5 summarizes our contribution and gives some prospects for future work.

2 Analytic Study of Two Weak-Perspective Views

2.1 The Camera Model

We assume that the scene object is stationary and that the camera translates by **T** and rotates by **R** around the object, and possibly zooms. A new affine coordinate frame associated with a second camera is given by the rows of $\mathbf{R}$ and the new origin lies at $-\mathbf{R}^\top \mathbf{T}$ thus a point in this second camera is given by the expression

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \frac{f'}{Z'_{\text{ave}}} \begin{bmatrix} X' \\ Y' \end{bmatrix}, \tag{1}$$

where $[X, Y, Z]^\top = \mathbf{R}[X', Y', Z']^\top + \mathbf{T}$, f' is the new focal length, and Z'_{ave} is the average distance to the object from the second camera.

Consider the equation $aX + bY + c = Z$ of a world plane $\mathcal{S}$. Then the two views of the coplanar scene are related by the affinity given by

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{M} \begin{bmatrix} x \\ y \end{bmatrix} + \mathbf{t}, \tag{2}$$

with

$$\mathbf{M} = s\frac{f'}{f} \begin{bmatrix} R_{1,1} + aR_{1,3} & R_{1,2} + bR_{1,3} \\ R_{2,1} + aR_{2,3} & R_{2,2} + bR_{2,3} \end{bmatrix}, \tag{3}$$

$$\mathbf{t} = -\frac{f'}{Z'_{\text{ave}}} \begin{bmatrix} R_{1,1} & R_{1,2} & R_{1,3} \\ R_{2,1} & R_{2,2} & R_{2,3} \end{bmatrix} \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} + c \begin{bmatrix} R_{1,3} \\ R_{2,3} \end{bmatrix}, \tag{4}$$

and where $s = Z_{\text{ave}}/Z'_{\text{ave}}$ is the scale factor that accounts for depth variation ($s > 1$ if the second camera approaches the scene object, and $s < 1$ if it departs from it), and $R_{i,j}$ are the elements of the rotation matrix $\mathbf{R}$.

A direction $\mathbf{v} = [x, y]^\top$ of the first image $\mathcal{R}$ is mapped by the above affinity to the direction $\mathbf{Mv}$ of the second image $\mathcal{R}'$. Since the affine references chosen in the two cameras match by the displacement, we can superpose the two images and it has sense to consider directions invariant by $\mathbf{M}$.

2.2 Recovery of the Epipolar Direction

Consider an orthonormal coordinate frame associated to the first image (for instance, normalized pixel coordinates, when aspect ratio and skew are known). The rotation matrix about the unit axis $[\cos\alpha, \sin\alpha, 0]^\top$ and angle ρ has the form

$$\mathbf{R} = \begin{bmatrix} (1-\cos\rho)\cos^2\alpha + \cos\rho & \cos\alpha\sin\alpha(1-\cos\rho) & \sin\alpha\sin\rho \\ \cos\alpha\sin\alpha(1-\cos\rho) & (1-\cos\rho)\sin^2\alpha + \cos\rho & -\cos\alpha\sin\rho \\ -\sin\alpha\sin\rho & \cos\alpha\sin\rho & \cos\rho \end{bmatrix}. \tag{5}$$

Hence, the matrix $\mathbf{M}$ is

$$\mathbf{M} = s\frac{f'}{f} \begin{bmatrix} \begin{matrix} (1-\cos\rho)\cos^2\alpha \\ +\cos\rho + a\sin\alpha\sin\rho \end{matrix} & \begin{matrix} \cos\alpha\sin\alpha(1-\cos\rho) \\ +b\sin\alpha\sin\rho \end{matrix} \\ & \\ \begin{matrix} \cos\alpha\sin\alpha(1-\cos\rho) \\ -a\cos\alpha\sin\rho \end{matrix} & \begin{matrix} (1-\cos\rho)\sin^2\alpha \\ +\cos\rho - b\cos\alpha\sin\rho \end{matrix} \end{bmatrix}, \tag{6}$$

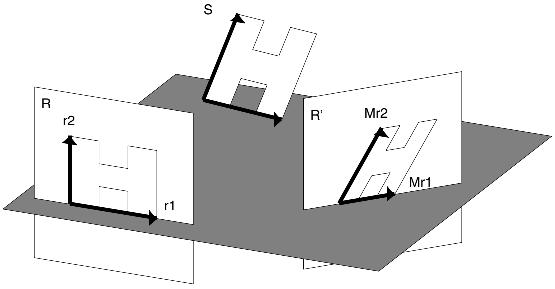


Fig. 1. Graphic illustration of Theorem 1. See text for details.

where $\mathbf{a} = [\cos\alpha, \sin\alpha]^\top$ is the direction of the rotation axis. The orthogonal vector $\mathbf{e} = [-\sin\alpha, \cos\alpha]^\top = \mathbf{a}^\perp$ is the epipolar direction. A straightforward computation shows that

$$\mathbf{Me} = s\frac{f'}{f}(\cos\rho + \sin\rho(a\sin\alpha - b\cos\alpha))\mathbf{e}\,, \tag{7}$$

thus giving an analytic proof of the following result:

Theorem 1. If the rigid motion between two weak-perspective cameras is assumed to be free of cyclorotation, then the epipolar direction $\mathbf{e}$ can be recovered as one of the two eigenvectors of the vectorial part $\mathbf{M}$ of the affinity that relates two views of a planar scene.

As a consequence, the direction $\mathbf{a} = \mathbf{e}^\perp$ of the axis of rotation can also be recovered.

Figure 1 illustrates the above result. Two views $\mathcal{R}$ and $\mathcal{R}'$ of a planar H-shaped object are shown, which are related by a rotation about an axis parallel to the image plane (i.e., free of cyclorotation). For simplicity of illustration, a basis $\{\mathbf{r_1}, \mathbf{r_2}\}$ is chosen aligned with the main axes of the H, and the axis of rotation is taken to be parallel to $\mathbf{r_2}$. Thus, the gray plane swept by $\mathbf{r_1}$ is left invariant by the rotation. Note, then, that the epipolar direction is that of $\mathbf{r_1}$ in $\mathcal{R}$ and that of $\mathbf{Mr_1}$ in $\mathcal{R}'$, and its perpendicular within each image is the direction of the rotation axis.

A geometric proof of Theorem 1 is included in [11]. Within the same geometrical framework, this result is generalized to the affine camera model leading to Theorem 2. Let us sketch the main ideas of this generalized result; the reader is referred to [11] for the details of the proof. The main advantage of this generalization is that, within the affine camera model, the projected target does not need to be centered in the image (assuming that the image center is a good approximation to the principal point). This enables us to handle a broader range of situations where the condition of small field of view is satisfied but the condition of being centered is relaxed. The affine camera model, which encloses the weak-perspective one, projects a scene point first under a fixed direction (which corresponds to a point $\overline{O}$ lying on the plane at infinity Π_∞) onto the average depth plane $\mathcal{R}^C$ (the plane parallel to the image plane $\mathcal{R}$ containing the centroid C of the scene object), and then perspectively from this fronto-parallel plane $\mathcal{R}^C$ onto the image $\mathcal{R}$. When $\overline{O}$ equals the direction O orthogonal to the image plane, the affine camera becomes a weak-perspective camera. By this projection procedure it is inferred that the affine camera, as well as the weak-perspective camera, preserves parallelism.

While in the weak-perspective camera model the improper optical center O is determined by the orientation of the image plane (i.e., O is the pole with respect to the absolute conic Ω of the improper line r of $\mathcal{R}$), in the affine camera model the improper optical center $\overline{O}$ may be any point in Π_∞. In fact, the direction of parallel projection, i.e., the improper optical center, depends on the position of the projected target within the image plane. This implies, on the one hand, that the same (pinhole) camera under affine viewing conditions can take two affine views with different improper optical centers (but keeping the same image plane). On the other hand, this also implies that, while the orientation of the image plane (and hence the improper optical center in case of a weak-perspective camera) is determined by the displacement performed by the camera, the improper optical center is not determined by the camera motion in the more general case of an affine camera. This is one of the reasons that makes the affine camera model more difficult to handle than the weak-perspective one.

Since the improper optical centers lie at infinity, the epipoles (of the first and second affine cameras) are also located at infinity in the image planes, i.e., the epipolar lines in both views are parallel. But, while in the weak-perspective cameras the epipoles coincide with the orthogonal direction (in the image plane) of the axis of rotation, in the general affine cameras the epipoles are no more related to this distinguished direction and, thus, a priori, they do not provide information about the rigid motion between the two affine cameras. This explains why most of the literature about the general affine camera model switches to the weak-perspective camera model when the question of inferring camera motion is addressed. Let us state the announced generalization result:

Theorem 2. Assume that the rigid motion between two affine cameras is free of cyclorotation and that the target projections are shifted (from the center of the image) along the direction orthogonal to the axis of rotation. Then the epipolar direction can be recovered as one of the two eigenvectors of the vectorial part $\mathbf{M}$ of the affinity that relates the two affine views of a planar scene.

2.3 Computing the Epipolar Direction from the Affinity Parameters

Fix any coordinate frame in the image (for instance pixel coordinates, since orthonormality is not required) and assume that the affinity that relates the two views has the expression

$$\mathbf{x}' = \mathbf{M}\mathbf{x} + \mathbf{t} = \begin{bmatrix} M_{1,1} & M_{1,2} \\ M_{2,1} & M_{2,2} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} . \tag{8}$$

In virtue of Theorem 1, the epipolar direction is one of the eigenvectors of $\mathbf{M}$. An eigenvector $[1, w]^\top$ of $\mathbf{M}$ satisfies the equation

$$M_{1,2}w^2 + (M_{1,1} - M_{2,2})w - M_{2,1} = 0\,. \tag{9}$$

If the motion is under the hypothesis of Theorem 1, then (9) must have two real solutions w_1, w_2, and the epipolar direction is $\mathbf{e} = [1, w_i]^\top$, for some $i \in \{1, 2\}$ (or $[0, 1]^\top$, in case $M_{1,2} = 0$).

3 Extracting the Affinity Parameters in Our Implementation

The affinity that relates two affine views is usually computed from a set of point matches. However, point matching is still one of the key bottlenecks in computer vision. In this work an active contour [12] is used instead. The active contour is fitted to a target object and the change of the active contour between different views is described by a shape vector deduced as follows. The contour is first represented as a parametric spline curve as it is common in Computer Graphics [13]. It has previously been shown [12] that the difference in control points $\mathbf{Q'} - \mathbf{Q}$ may be written as a linear combination of six vectors. Therefore, using matrix notation,

$$\mathbf{Q'} - \mathbf{Q} = \mathbf{WS}\,, \tag{10}$$

where

$$\mathbf{W} = \left(\begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix}, \begin{bmatrix} \mathbf{Q^x} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{0} \\ \mathbf{Q^y} \end{bmatrix}, \begin{bmatrix} \mathbf{0} \\ \mathbf{Q^x} \end{bmatrix}, \begin{bmatrix} \mathbf{Q^y} \\ \mathbf{0} \end{bmatrix} \right)\,, \tag{11}$$

and $\mathbf{S}$ is a vector with the six parameters of the linear combination, the shape vector

$$\mathbf{S} = [t_x, t_y, M_{1,1} - 1, M_{2,2} - 1, M_{2,1}, M_{1,2}]^\top\,, \tag{12}$$

which encodes the relation between different affine views of the planar contour.

Note that the dimension of the shape vector can be reduced if robot motion is constrained, for instance to lie on a plane [14].

Once the compact representation of the contour in terms of control points and knots is obtained, a Kalman filter is used to track the contour along the sequence [12], and the shape vector is updated at each frame.

In previous works [15,16], the continuously updated shape vector was used to estimate robot egomotion in practice, provided data from other sensors (such as an inclinometer) or scene information (such as depth) were supplied. Here we focus on the extraction of epipolar direction from the shape vectors of just two views, and the analysis of the attainable accuracy in the different possible working conditions.

4 Experimentation

Two sets of experiments were performed to evaluate the accuracy of the proposed method. The first set uses synthetic image sequences generated by simulating camera motion and computing projections under a full perspective camera model. Using this set, the sensitivity of the proposed algorithm to perspectivity effects is assessed by changing the distance of the target to the camera. A complete study involving the relaxation of all weak-perspective hypotheses can be found in [11].

The affine epipolar geometry is usually estimated using the Gold Standard algorithm [5]. This technique requires image correspondences of at least 4 non-coplanar points. Using also our synthetic experimental testbed, we show the effects of approaching coplanarity for this configuration, and compare the results with those of our method.

The second set of experiments uses real images taken by a robot arm moving along a calibrated path, showing the performance of the approach under realistic imaging conditions. In this setting, a comparison with the Gold Standard algorithm is also provided.

4.1 Simulations

When synthetic images are generated using an affine camera model (i.e., assuming perfect weak-perspective conditions), the epipolar direction is exactly recovered with the proposed method. However, we would like to assess the validity of the method under more general conditions. To this end, we generate the test set of synthetic images using a full perspective camera model. Then, of course, perspectivity effects affect the recovery of the epipolar direction in the ways that will be analysed in the following.

In the first experiment we analyse how a decrement of the distance Z_{ave} from the camera to the target affects the computation of the epipolar direction. Decreasing the distance enlarges perspective effects, and consequently, should increase the error in epipolar direction recovery. For this experiment we consider distances of $500, 750, 1000, 1250, 1500, 1750$ and $2000mm$. The smallest of these, $500mm$, corresponds to an extreme situation for the weak-perspective model, in which important unmodelled distortions in the projected control polygon are present. For larger depth values, the affine conditions are better satisfied, thus reducing the error, as shown in Figure 2. It is worth noting that even under these unfavourable conditions the recovery error stays below 0.6°.

The effects of relaxing other assumptions, such as lateral translations leading to uncentered targets, introducing depth relief, or having cyclorotation have also been explored and the results are given in [11], where the sensitivity to contour shape is also analysed.

Next we describe a comparison with a standard technique for computing the affine epipolar geometry, namely the Gold Standard (GS) algorithm [5]. This algorithm, contrary to our procedure, needs non-coplanar point correspondences in order to compute the maximum likelihood estimate of the affine fundamental matrix. While in theory, only four non-coplanar points would suffice for computing the affine epipolar geometry using the GS algorithm, its performance is affected by the amount of non-coplanar information provided, both in terms of de pth range and in the number of points used. The idea is to establish experimen-

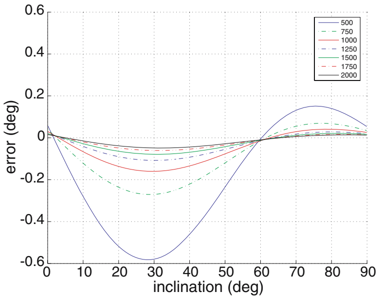


Fig. 2. Effects of relaxing one of the weak-perspective conditions by varying the distance from the camera to the target. The camera rotation is of 40° about an axis on the target with inclination of 45°.

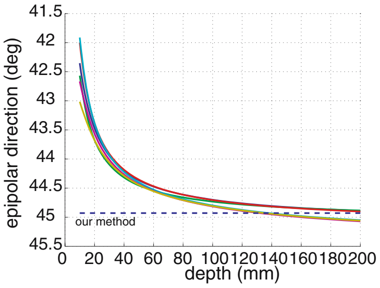
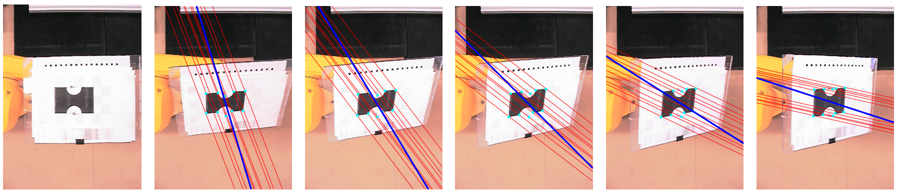


Fig. 3. Epipolar direction computed with the GS algorithm in the case of 2,4,...,12 out-of-plane points (a curve for each number) placed at increasing depths (in abscissae) above the H-shaped contour

tally the amount of depth information required by GS algorithm for it to provide equivalent epipolar direction recovery results to our procedure.

To this end, we set first an experiment in which we add a range from two to twelve extra points to the H-shaped contour, varying their distance with respect to the contour plane. Camera parameters are fixed at: 500 mm distance to target and a focal distance of 767 $pixels$. As before, camera motion is achieved via a rotation of 40° about an axis placed at an orientation of 45° on the target plane. The results are shown in Figure 3. It can be seen how as the depth of these points is increased, the error in the computation of the epipolar direction decreases. Moreover, it turns out that the number and xy location of these points have little effect in the computation of the epipolar direction. The figure contains plots of the resulting errors in the computation of the affine epipolar direction with the GS algorithm for different numbers of out-of-plane points, and a threshold indicating the error in the recovery of the epipolar direction using



(a) Init (b) 15° (c) 30° (d) 45° (e) 60° (f) 75°

Fig. 4. The first experiment with real images entails pairs of views consisting of the initial one plus each of the other five, corresponding to camera rotations of 40° about an axis on the target with inclinations sampled at intervals of 15°. The epipolar direction computed by the proposed technique is displayed as a line passing through the target center, while the thin lines are the epipolar lines obtained with GS.

Table 1. Mean and standard deviation in degrees of the epipolar direction computed by the proposed technique and the GS algorithm from real images

epipolar direction	-15	-30	-45	-60	-75
$\bar{\theta}$	-16.63	-31.01	-45.00	-57.63	-72.04
σ	0.14	0.09	0.14	0.19	0.13
θ_{GS}	-18.53	-34.25	-49.46	-62.53	-76.36

our proposed technique under the same experimental conditions (the additional points out of the contour plane are evidently not used in this case). As shown in the figure, for the given experimental conditions, the results of our technique are comparable to those of the Gold Standard algorithm when the extra points are placed roughly at a distance equal to the target size (120 *mm* in our case).

Note the importance of parallax in the computation of the affine fundamental matrix with the Gold Standard algorithm. As the target points approach coplanarity, the parallax vector, which determines the epipolar direction, is monotonically reduced in length. Consequently, the accuracy of the line direction is also reduced, and the covariance of the estimated affine fundamental matrix increases. This situation does not occur in our procedure, as it has been devised precisely to compute the affine epipolar direction from two views of a plane.

4.2 Experiments Using Real Images

We present now results on image sequences in a controlled setting of our technique for computing the affine epipolar direction from pairs of views of a plane only. The goal of this work is not tracking, but computing the affinity from an active contour deformation, and using it to estimate the epipolar direction induced by the two views. To this end, we facilitate the tracking phase by moving a simple target placed on a manipulator end-effector, and focus on evaluating the accuracy of the direction recovered in different situations, compared to robot motion ground truth.

The experimentation setup consists of a Stäubli RX60 manipulator holding the target pattern on its end-effector. This target is a planar artificial H-shaped figure with corners and curved edges, which can be easily tracked with our active contour tracker. We are interested in using such setup in order to obtain a precise ground truth for the experiment. The initial distance from camera to target has had to be set to 500 *mm*. This corresponds to the extreme case discussed in Section 4.1, Fig. 2, and, therefore, we are testing the proposed approach under relaxed weak-perspective conditions. The acquired images have evident perspective effects, as shown in Figures 4 and 5, which make our algorithm work under extreme conditions. In order to provide depth information to the GS algorithm, the endpoints of two 20 *mm* screws placed at both sides of the contour are used as matching features in junction with the eight corners of the contour. Note that these are also extreme conditions for the GS algorithm to work, since

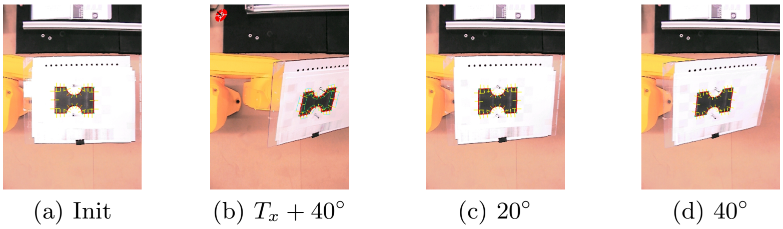

(a) Init (b) $T_x + 40°$ (c) 20° (d) 40°

Fig. 5. Experiments with real images further relaxing weak-perspective conditions. The first sequence, entailing an uncentered target, starts at (a) and ends at (b). The next one departing from a non-frontoparallel target position starts at (c) and ends at (d).

Table 2. Mean and standard deviation of the epipolar direction computed over real images when weak-perspective conditions are further relaxed

Frames	$\bar{\theta}$	σ	θ_{GS}
Not Centered	-34.65	0.13	-56.29
Not Frontoparallel	-43.89	0.09	-49.78

very little depth information is provided: only two out-of-plane points. Thus, due to the setup we currently have, we are comparing both algorithms at the limit of their respective working conditions.

The first experiment entails camera motion induced by a rotation of 40° about an axis on the target at various inclination angles sampled at intervals of 15°. This, thus, relates to Fig. 2 with distance equal to 500 *mm*. Starting from the fronto-parallel position shown in Figure 4(a), the contour is tracked to each of the final views shown in the remaining frames of the figure. The epipolar direction computed by the proposed algorithm in each case is displayed as a line passing through the target center. Thin lines passing through the points correspond to the epipolar direction computed with the GS algorithm.

Table 1 presents the numerical values obtained in the computation of the epipolar direction. Standard deviation is computed by acquiring 300 images in the final position, estimating the shape vectors and then computing the corresponding epipolar directions. Note that the standard deviations are all very similar, and the mean values deviate more from ground truth as the angle departs from the 45° inclination. This should be interpreted in the light of Fig. 2 as meaning that the tracker amplifies the recovery error due to perspectivity effects unmodelled by the weak-perspective camera. Consequently, under true weak-perspective conditions, the errors should be much lower as indicated by the shrinking of the error curves in Fig. 2 when the distance Z_{ave} from the camera to the target increases. Results using the GS algorithm are sightly worse than those obtained with the proposed algorithm. This is due to perspective effects as well as to the poor depth information provided with the point matches used.

Two additional sequences were analyzed after further relaxing weak-perspective conditions. The first such sequence, labelled "Not centered", starts at the fronto-parallel initial position (Fig. 5(a)) and finishes at an uncentered position, after a translation of 100 mm along the x axis of the robot coordinate frame and a rotation of 40° about an axis at 45° inclination (Fig. 5(b)). Consistent with our simulated results [11], this lateral camera translation is by far the violation of weak-perspective conditions that has the most pervasive effect on the computation of the epipolar direction. See the numbers in Table 2, first row, which is far from the motion assumption of Theorem 2. This pervasive effect appears also in the computation with the GS algorithm, yielding the largest error in the experiments.

The second experiment, labelled "Not Frontoparallel", corresponds to the same rotation described above, but the initial frame is not frontoparallel. The sequence starts with the target already rotated 20° as shown in Fig. 5(c) and, after a further rotation of 20°, finishes at 40° (Fig. 5(d)), all rotations about an axis at 45° inclination as before. Observe that the result is only a bit worse than that of the initial experiment, but with a similar standard deviation. The result with the GS algorithm here is similar as before.

5 Conclusions

The recovery of camera motion and scene structure from uncalibrated image sequences has received a lot of attention lately due to its numerous applications, which range from robot localization and navigation, to virtual reality and archeology, to name just a few. Most works rely on detecting a set of non-coplanar points in the scene and matching their projections on the different views. In this paper we have departed from this main stream, by dealing with a less informative situation, namely features lying on a plane, and recurring to contour tracking instead of point matching.

Our main result is that, under weak-perspective conditions and assuming a camera motion free of cyclorotation, the epipolar direction can be recovered from the affinity relating two views of a planar scene.

Synthetic images were used to evaluate the results in a noise-controlled environment, and then to compare the accuracy of our method with that of the Gold Standard algorithm, which relying on matches of non-coplanar points falls in the main stream mentioned above.

The outcome of the comparison has been very encouraging, since with less scene information (only from a plane) and with a much simpler processing (solving a single second-order equation), we are able to obtain the epipolar direction with similar accuracy. It is worth reminding, however, that our method is less general in that it requires a camera motion free of cyclorotation.

The second experimental set consisted of image sequences that were used to validate the proposed approach under real imaging conditions. Note that the objective of the paper is to show what can be obtained from the affine deformation of two views of a contour, and not to validate the robustness of the contour tracker used. For this reason, simple and well-calibrated image sequences were used in order to have a good basis for ground truth comparison.

Future work will include an error analysis that involves positional errors on the contours due to the image acquisition process. Moreover, we will try to unravel under what circumstances additional information on camera motion and scene structure can be recovered from two (or more) uncalibrated views of a planar object. Along the same line, we will tackle the recovery of the orientation of the scene plane, as well as what occurs in degenerate situations in which such orientation is the same as that of the image plane, or when both planes have a common direction.

References

1. Beardsley, P.A., Zisserman, A., Murray, D.W.: Sequential updating of projective and affine structure from motion. Intl. J. of Computer Vision **23** (1997) 235–259
2. McLauchlan, P.F., Murray, D.W.: A unifying framework for structure and motion recovery from image sequences. In: Proc. Intl. Conf. on Computer Vision. (1995) 314–320
3. Koenderink, J., van Doorn, A.J.: Affine structure from motion. J. Opt. Soc. Am. A **8** (1991) 377–385
4. Shapiro, L., Zisserman, A., Brady, M.: 3d motion recovery via affine epipolar geometry. Intl. J. of Computer Vision **16** (1995) 147–182
5. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision (Second Edition). Cambridge University Press (2004)
6. Sturm, P., Maybank, S.J.: On plane-based camera calibration: a general algorithm, singularities, applications. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. Volume 1. (1999) 432–437
7. Demirdjian, D., Zisserman, A., Horaud, R.: Stereo autocalibration from one plane. In: Proc. 6th European Conf. on Computer Vision. (2000) 625–639
8. Malis, E., Cipolla, R.: Camera self-calibration from unknown planar structures enforcing the multiview constraints between collineations. IEEE Trans. on Pattern Analysis and Machine Intelligence **24** (2002) 1268–1272
9. Bartoli, A., Sturm, P., Horaud, R.: Structure and motion from two uncalibrated views using points on planes. In: Proc. 3rd. Intl. Conf. on 3D Digital Imaging and Modeling, Canada (2001) 83–90
10. Kaminski, J.Y., Shashua, A.: On calibration and reconstruction from planar curves. In: Proc. European Conf. on Computer Vision. (2000) 678–694
11. Alberich-Carramiñana, M., Alenyà, G., Andrade-Cetto, J., Martínez, E., Torras, C.: Affine epipolar direction from two views of a planar contour. Technical Report IRI-DT-2005/03, Institute of Robotics (IRI) (2005)
12. Blake, A., Isard, M.: Active Contours. Springer (1998)
13. Foley, J., van Dam, A., Feiner, S., Hughes, F.: Computer Graphics. Principles and Practice. Addison-Wesley Publishing Company (1996)
14. Alenyà, G., Martínez, E., Torras, C.: Fusing visual and inertial sensing to recover robot egomotion. Journal of Robotics Systems **21** (2004) 23–32
15. Martínez, E., Torras, C.: Qualitative vision for the guidance of legged robots in unstructured environments. Pattern Recognition **34** (2001) 1585–1599
16. Martínez, E., Torras, C.: Contour-based 3d motion recovery while zooming. Robotics and Autonomous Systems **44** (2003) 219–227

Orientation Invariant Features for Multiclass Object Recognition*

Michael Villamizar[1], Alberto Sanfeliu[1], and Juan Andrade-Cetto[2]

[1] Institut de Robòtica i Informàtica Industrial, UPC-CSIC
Llorens Artigas 4-6, 08028 Barcelona, Spain
{mvillami, sanfeliu}@iri.upc.edu
[2] Computer Vision Center, Universitat Autònoma de Barcelona
Edifici O, Campus UAB, 08193 Bellaterra, Spain
cetto@cvc.uab.es

Abstract. We present a framework for object recognition based on simple scale and orientation invariant local features that when combined with a hierarchical multiclass boosting mechanism produce robust classifiers for a limited number of object classes in cluttered backgrounds. The system extracts the most relevant features from a set of training samples and builds a hierarchical structure of them. By focusing on those features common to all trained objects, and also searching for those features particular to a reduced number of classes, and eventually, to each object class. To allow for efficient rotation invariance, we propose the use of non-Gaussian steerable filters, together with an Orientation Integral Image for a speedy computation of local orientation.

1 Introduction

Object detection is a fundamental issue in most computer vision tasks; particularly, in applications that require object recognition. Early approaches to object recognition are based on the search for matches between user-generated geometrical object models and image features. To overcome the need of such models, appearance-based object recognition gained popularity in the past two decades using dimensionality reduction techniques such as PCAs for whole-image matching. Unfortunately, appearance based matching as such, is prone to fail in situations with modest occlusions or under varying backgrounds. Lately, a new paradigm for object recognition has appeared based on the matching of geometrical as well as appearance local features. The most popular of these, perhaps, the SIFT descriptor [1].

* Financial support to M. Villamizar and A. Sanfeliu comes from the EURON Network Robot Systems Research Atelier NoE-507728, and the Spanish Ministry of Education and Science project NAVROB DPI 2004-05414. J. Andrade-Cetto is a Juan de la Cierva Postdoctoral Fellow of the Spanish Ministry of Education and Science under project TIC2003-09291, and is also funded in part by the EU PACO-PLUS project FP6-2004-IST-4-27657. The authors belong to the Artificial Vision and Intelligent Systems Group funded in part by the Catalan Research Commission.

J.F. Martínez-Trinidad et al. (Eds.): CIARP 2006, LNCS 4225, pp. 655–664, 2006.

Instead of using general saliency rules for feature selection as in the case of the SIFT descriptor, the use of boosting techniques for feature selection has proven beneficial in choosing the most discriminant geometric and appearance features from training sets. Despite their power in achieving accurate recognition from trained data, early boosting mechanisms such as [2], were tailored to single class object recognition, and are not suitable for multiclass object recognition given the large amount of features that need to be trained independently for each object class. Lately however, there have been some extensions to the general idea of classfication with boosting that allow the combined training of multiple classes [3,4]. In the computer vision doamin, Torralba *et al.* [5] proposed an extension to one such boosting algorithm (gentleboost), with the purpose of sharing features across multiples object classes so as to reduce the total number of classifiers. They called it JointBoost, and in this approach, all object classes are trained jointly, and for each possible subset of classes ($2^n - 1$ excluding the empty set), the most useful feature is selected to distinguish that subset from the background class. The process is repeated until the overall classification error reaches a minimum, or until a limit on the number of classifiers is achieved.

The type of weak classifier features used in [5] are very simple template matching masks, that would presumibly fail if sample objects are to be found at different orientations than as trained. In this work we investigate on the use of similar multiclass feature selection, but with keen interest in fast computation of orientation invariant weak classifiers [6] for multiclass rotation invariant object recognition.

In [2], Viola introduced the integral image for very fast feature evaluation. Once computed, an integral image allows the computation of Haar-like features [7] at any location or scale in real time. Unfortunately, such system is not invariant to object rotation or occlusions. Other recognition systems that might work well in cluttered scenes are based on the computation of multi-scale local features such as the previously mentioned SIFT descriptor [1]. One key idea behind the SIFT descriptor is that it incorporates canonical orientation values for each keypoint. Thus, allowing scale and rotation invariance during recognition. Even when a large number of SIFT features can be computed in real time for one single image, their correct pairing between sample and test images is performed via nearest neighbor search and generalized Hough transform voting, followed by the solution of the affine relation between views; which might end up to be a time consuming process.

Yokono and Poggio [8,9] settle for Harris corners at various levels of resolution as interest points, and from these, they select as object features those that are most robust to Gaussian derivative filters under rotation and scaling. As Gaussian derivatives are not rotation invariant, they use steerable filters [10] to steer all the features responses according to the local gradient orientation around the interest point. In the recognition phase, the system still requires local feature matching, and iterates over all matching pairs, in groups of 6, searching for the best matching homography, using RANSAC for outlier removal. Unfortunately, the time complexity or performance of their approach was not reported.

In [6] we realized that filter response to Haar masks can be not only be computed efficiently with an integral image scheme; but also, that such masks can be approximately rotated with some simplifications of the Gaussian steerable filter. Thus, allowing for fast computation of rotation invariant filter responses as week classifiers.

In this paper, we incorporate these two ideas, multiclass boosting, and rotation invariance, for the selection of joint and specific local features to construct a hierarchical structure that allow recognizing multiples objects independently of position, scale and orientation with a reduced set of features. In our system, keypoints are chosen as those regions in the image that have the most discriminant response under convolution with a set of wavelet basis functions at several scales and orientations. Section 2 explains how the most relevant features are selected and combined to classify multiples objects. The selection is based on JointBoost, in which a hierarchical structure is composed by sets of joint and specific classifiers. A linear combination of these weak classifiers produces a strong classifier for each object class, which is used for detection. Rotation invariance is achieved by filtering with oriented basis functions. Filter rotation is efficiently computed with the aid of a steerable filter, that is, as the linear combination of basis filters, as indicated in Section 3.

During the recognition phase, sample image regions must be rotated to a trained canonical orientation, prior to feature matching. Such orientation is dictated by the peak on a histogram of gradient orientations, depicted in Section 4. Section 5 explains our proposed Orientation Integral Image for the speed of kernel orientation computation, and Section 6 presents some experiments.

2 Feature Selection

The set of local features that best discriminates an object is obtained by convolving positive sample images with a simplified set of wavelet basis function operators [7] at different scales and orientations. These filters have spatial orientation selectivity as well as frequency selectivity, and produce features that capture the contrast between regions representing points, edges, and strips, and have high response along for example, contours. The set of operators used is shown in Figure 1. Filter response is equivalent to the difference in intensity in the original image between the dark and light regions dictated by the operator. Figure 1 d) exemplifies how an object can be represented by a small set of the most useful local features.

Convolving these operators at any desired orientation is performed by steering the filter (Section 3), and fast convolution over any region of the entire image is efficiently obtained using an integral image (Section 5).

Feature selection is performed as in JointBoost [5], choosing one at a time, from the $2n - 1$ subsets of the classes $c = 1...n$ (empty set excluded), the weak classifier $h(I, s)$ that best discriminates any subset s from the background class

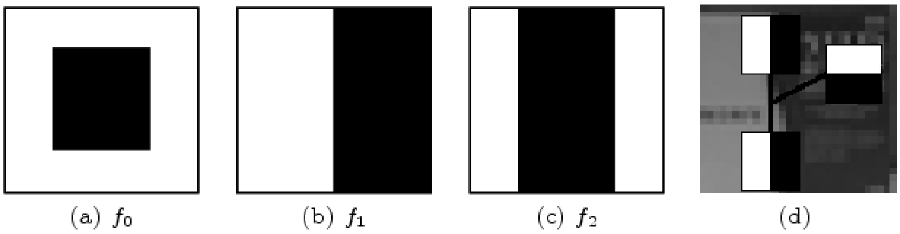


Fig. 1. Simplified wavelet basis function set. a) center-surround b) edge, and c) line; and d) object local features.

(lowest classification error). The weak classifier is defined by the parameters filter type, size, location, orientation and threshold, taking the binary decission value

$$h(I,s) = \begin{cases} 1 & : \quad I * f > t \\ 0 & : \quad \text{otherwise} \end{cases} \tag{1}$$

where I is a training sample image of class c in the subset s, f is the filter being tested, with all its parameters, $*$ indicates the convolution operation, and t is the filter response threshold.

At each iteration during the training phase, the algorithm must find for all of the $2n - 1$ subsets, the weak classifier that best discriminates that subset from the background class by minimizing the squared error over weighted samples of all classes in that subset

$$J_{wse} = \sum_{c=1}^{n} \sum_{s=1}^{m} w_i^c (z_i^c - h(I,s))^2 \tag{2}$$

where z_i^c and w_i^c are the membership label and weight of the sample i for class c respectively, and m the total number of training samples. The algorithm also updates sets of weights over the training samples. The number of sets corresponds with the number of classes to learn. Initially, all weights are set equally, but on each round, the weights of missclassified samples are increased so that the algorithm is forced to focus on such hard samples in the training set the previously chosen classifiers missed. Finally, choosing the weak classifier for the subset that had the minimum squared error J, and iteratively adding it to the Strong Classifier for every class c in s, $H(I,c)$,

$$H(I,c) := H(I,c) + h(I,s) \tag{3}$$

Scale invariance is obtained by iterating also over scaled filters within the classifier H. Scaling of the filters can be performed in constant time for a previously computed integral image.

3 Steerable Filters

In order to achieve orientation invariance, the local filters must be rotated previous to convolution. A good alternative is to compute these rotations with steerable filters [10], or with its complex version [11]. A steerable filter is a rotated filter comprised of a linear combination of a set of oriented basis filters

$$I * f(\theta) = \sum^{n} k_i(\theta) I * f(\theta_i) \; , \tag{4}$$

where $f(\theta_i)$ are the oriented basis filters, and k_i are the coefficients of the bases.

Consider for example, the Gaussian function $G(u, v) = e^{-(u^2+v^2)}$, and its first and second order derivative filters $G'_u = -2ue^{-(u^2+v^2)}$ and $G''_u = (4u^2 - 2)e^{-(u^2+v^2)}$. These filters can be re-oriented as a linear combination of filter bases. The size of the basis is one more than the derivative order.

Consequently. the first order derivative of our Gaussian function at any direction θ is

$$G'_\theta = \cos\theta G'_u + \sin\theta G'_v \; , \tag{5}$$

and, the steered 2nd order Gaussian filter can be obtained with

$$G''_\theta = \sum_{i=1}^{3} k_i(\theta) G''_{\theta_i} \tag{6}$$

with $k_i(\theta) = \frac{1}{3}(1 + 2\cos(\theta - \theta_i))$; and G''_{θ_i} precomputed second order derivative kernels at $\theta_1 = 0$, $\theta_2 = \frac{\pi}{3}$, and $\theta_3 = \frac{2\pi}{3}$. See Figure 2.

Convolving with Gaussian kernels is a time consuming process. Instead, we propose in [6] to approximate such filter response by convolving with the Haar basis with the objective of using the integral image. Thus, we approximate the oriented first derivative response with

$$I * f_1(\theta) = \cos\theta I * f_1(0) + \sin\theta I * f_1(\tfrac{\pi}{2}) \; . \tag{7}$$

and in the same sense, the filtering with our line detector at any orientation θ is obtained with

$$I * f_2(\theta) = \sum_{i=1}^{3} k_i(\theta) I * f_2(\theta_i) \; . \tag{8}$$

The similarity of the response between the Gaussian and the Haar filters allows us to use the later basis instead as weak classifiers for the detection of points, edges, and lines; just as the Gaussian filters do. The main benefit of the approach is in speed of computation. While convolution with a Gaussian kernel takes time O(n) the size of the kernel, convolution with the oriented Haar basis can be computed in constant time using an integral image representation. Figure 3 shows some results.

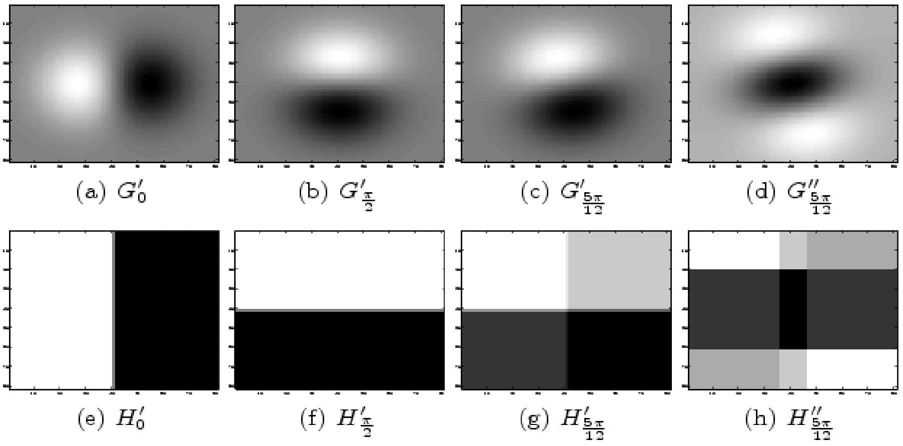

(a) G'_0
(b) $G'_{\frac{\pi}{2}}$
(c) $G'_{\frac{5\pi}{12}}$
(d) $G''_{\frac{5\pi}{12}}$
(e) H'_0
(f) $H'_{\frac{\pi}{2}}$
(g) $H'_{\frac{5\pi}{12}}$
(h) $H''_{\frac{5\pi}{12}}$

Fig. 2. First and second order steerable filters. (a-b) Gaussian basis, (c-d) Gaussian oriented filters, (e-f) Haar basis, (g-h) Haar oriented filters.

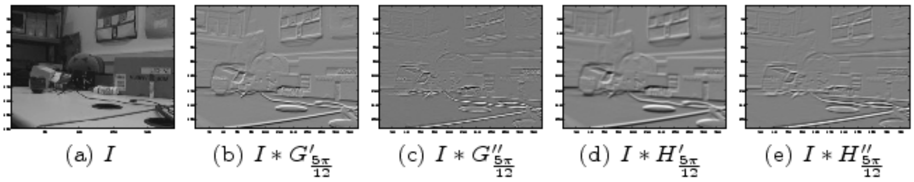

(a) I
(b) $I * G'_{\frac{5\pi}{12}}$
(c) $I * G''_{\frac{5\pi}{12}}$
(d) $I * H'_{\frac{5\pi}{12}}$
(e) $I * H''_{\frac{5\pi}{12}}$

Fig. 3. Filter responses. (a) original image, (b-e) filter responses.

4 Local Orientation

Consider a training session has produced a constellation H of local features h as the one shown in Figure 4. Now, the objective is to test for multiple positions and scales in each new image, whether such constellation passes the test H or not. Instead of trying every possible orientation of our constellation, we chose to store the canonical orientation θ_0 of H from a reference training image block, and to compare it with the orientation θ of each image block being tested. The difference between the two indicates the amount we must re-orient the entire feature set before the test H is performed.

On way to compute block image orientation is with ratio of first derivative Gaussians G'_u and G'_v [9], $\tan\theta = \frac{I*G'_v}{I*G'_u}$. Another technique, more robust to partial occlusions, is to use the mode of the local gradient orientation histogram (see Figure 4 c-d), for which it is necessary to compute gradient orientations pixel by pixel, instead of a region convolution as in the previous case.

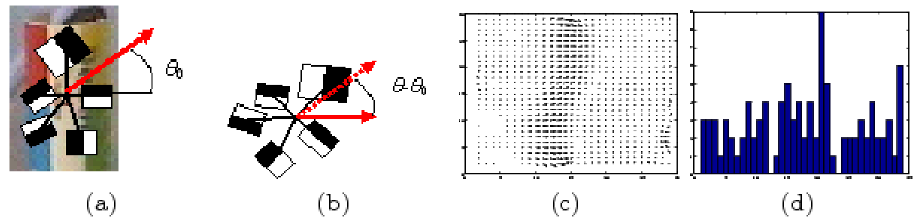


Fig. 4. Local orientation a) canonical orientation, b) rotated constellation, c) image gradients, b) gradient orientation histogram

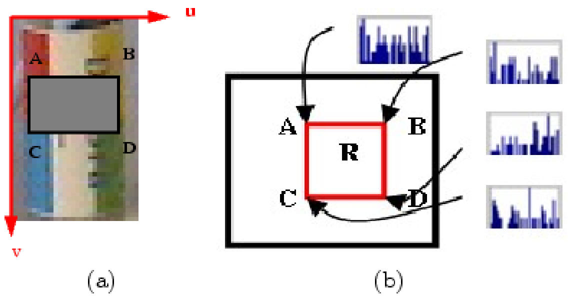


Fig. 5. Integral Images, a) Integral Image b) Orientation Integral Image

5 The Local Orientation Integral Image

An integral image is a representation of the image that allows a fast computation of features because it does not work directly with the original image intensities. Instead, it works over an incrementally built image that adds feature values along rows and columns. Once computed this image representation, any one of the local features (weak classifiers) can be computed at any location and scale in constant time.

In its most simple form, the value of the integral image M at coordinates u, v contains the sum of pixels values above and to the left of u, v, inclusive.

$$M(u,v) = \sum_{i \leq u, j \leq v} I(i,j) \tag{9}$$

Then, it is possible to compute for example, the sum of intensity values in a rectangular region simply by adding and subtracting the cumulative intensities at its four corners in the integral image (Figure 5a). Then, the response from the Haar-filters can be calculated in a fast way independently of size or location.

$$\text{Area} = A + D - B - C \tag{10}$$

Extending the idea of having cumulative data at each pixel in the Integral Image, we decide to store in it orientation histogram data instead of intensity sums. Once constructed this orientation integral image, it is possible to compute a local orientation histogram for any given rectangular area within an image in constant time. see Figure 5b.

$$\begin{aligned}\text{Histogram(Area)} &= \text{Histogram}(A) + \text{Histogram}(D) \\ &\quad - \text{Histogram}(B) - \text{Histogram}(C) \qquad (11)\end{aligned}$$

6 Experiments

In this communication we report on initial recognition results for a limited number of objects in gray scale images. The training set had 100 images for each class, and 500 negatives or background images. These negatives images were extracted from exterior and interior scenes. The positive class images used for training presented some small translation, orientation, and scale, as shown in Figure 6.

Figure 7 a) and b) show examples of extracted feature constellation for each object class. Each one is composed by 8 weak classifiers (Haar-like features), with 4 of them common to both classes, and the remaining 4 specific to each class.

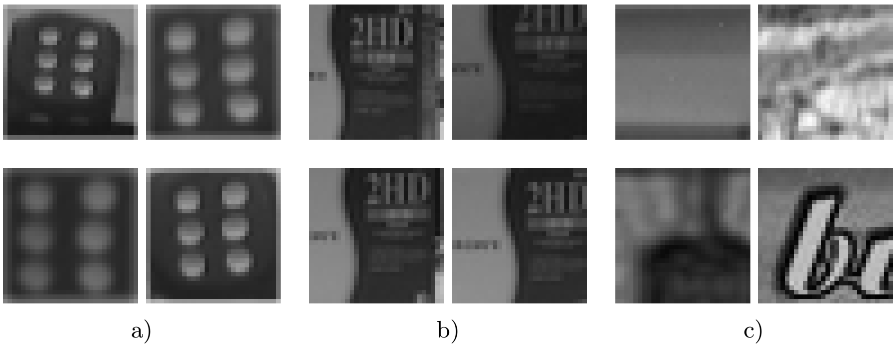


Fig. 6. Training object classes. a) dice images, b) CD box images, and c) background images.

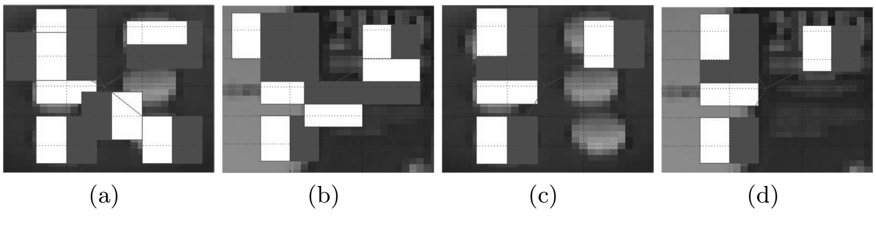
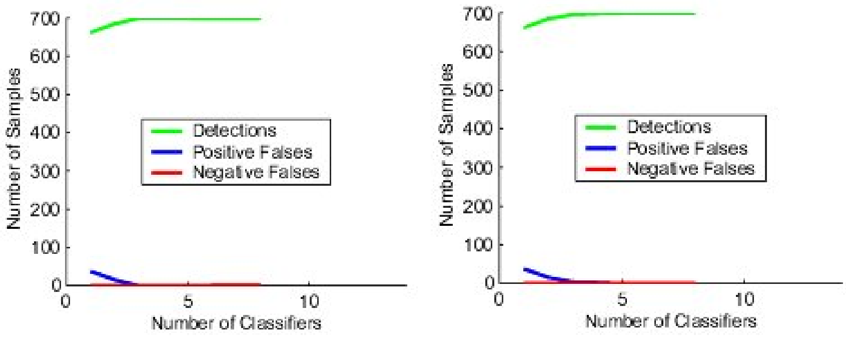


Fig. 7. Constellations. a) dice constellation b) CD box constellation (c-d) joint classifiers.



(a) (b)

Fig. 8. Training performance. a) dice b) CD box.

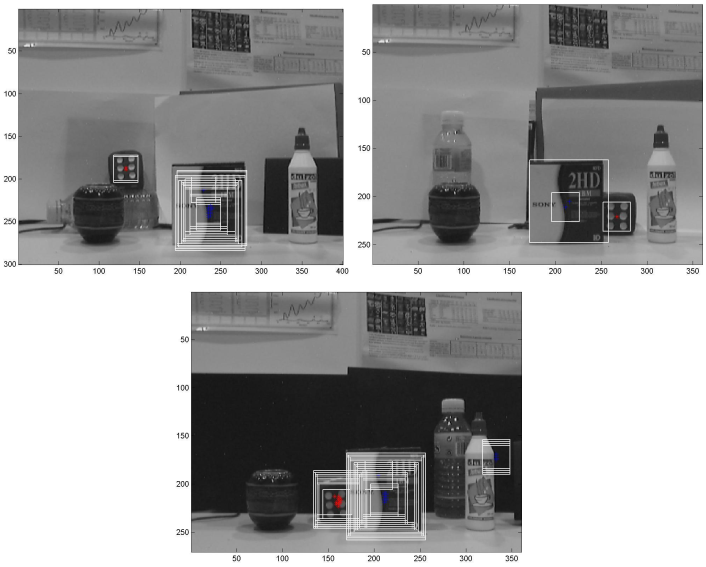


Fig. 9. Examples of correct detection of classifiers trained jointly (dice and Cd box). The last image shows also under what circumpstances a false detection might occur.

Thus, producing a hierarchical structure of weak classifiers. Frames c) and d) show only those four classifiers that are common to both classes. They capture simmilar local information in both classes, separating them from the background set, without the need to be class specific.

The Strong Classifiers can be expressed as the combination of joint and specific weak classifiers. Consider the dice to be class 1, the CD box to be class 2, and c_{12} the set of training samples containing either one or both objects. Then

$$H(I, c_1) = \sum h(I, c_{12}) + \sum h(I, c_1) \tag{12}$$

$$H(I, c_2) = \sum h(I, c_{12}) + \sum h(I, c_2) \tag{13}$$

The training curves are shown in Figure 8.They illustrate how the correct classification of the training set is achieved. Some results in detection process over a image sequence are visualized in Figure 9.

7 Conclusions

In this paper we have introduced a hierarchical feature selection structure that reduce the total number of weak classifiers needed to detect multiples object classes. With this method the system finds common features among objects and generalizes the detection problem.

Our approach is based on boosting over a set of simple local features. In contrast to previous approaches, and to efficiently cope with orientation changes, we propose the use of Haar basis functions and a new orientation integral image for a speedy computation of local orientation.

References

1. Lowe, D.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision **60** (2004) 91–110
2. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. 15th IEEE Conf. Comput. Vision Pattern Recog., Kauai (2001) 511–518
3. Li, L.: Multiclass boosting with repartitioning. In: Proc. 23rd Int. Conf. Machine Learning, Pittsburgh (2006) To appear.
4. Eibl, G., Pfeiffer, K.P.: Multiclass boosting for weak classifiers. J. Mach. Learn. Res. **6** (2005) 189–210
5. Torralba, A., Murphy, K., Freeman, W.: Sharing features: efficient boosting procedures for multiclass object detection. In: Proc. 18th IEEE Conf. Comput. Vision Pattern Recog., Washington (2004) 762–769
6. Villamizar, M., Sanfeliu, A., Andrade-Cetto, J.: Computation of rotation local invariant features using the integral image for real time object detection. In: Proc. 18th IAPR Int. Conf. Pattern Recog., Hong Kong, IEEE Comp. Soc. (2006) To appear.
7. Papageorgiou, C.P., Oren, M., Poggio, T.: A general framework for object detection. In: Proc. IEEE Int. Conf. Comput. Vision, Bombay (1998) 555
8. Yokono, J., Poggio, T.: Oriented filters for object recognition: An empirical study. In: Proc. 6th IEEE Int. Conf. Automatic Face Gesture Recog., Seoul (2004) 755–760
9. Yokono, J., Poggio, T.: Rotation invariant object recognition from one training example. Technical Report 2004-010, MIT AI Lab. (2004)
10. Freeman, W.T., Adelson, E.H.: The design and use of steerable filters. IEEE Trans. Pattern Anal. Machine Intell. **13** (1991) 891–906
11. Schaffalitzky, F., Zisserman, A.: Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?". In: Proc. 7th European Conf. Comput. Vision, Copenhagen, Springer-Verlag (2002) 414–431

Active Control for Single Camera SLAM

Teresa Vidal-Calleja
Institut de Robòtica i Informàtica Industrial, CSIC-UPC
Llorens Artigas 4-6, Barcelona 08028, Spain.
tvidal@iri.upc.edu

Andrew J. Davison
Department of Computing, Imperial College London
180 Queen's Gate, London SW7 2AZ, UK.
ajd@doc.ic.ac.uk

Juan Andrade-Cetto
Centre de Visiò per Computador, UAB
Edifici O, Campus UAB, Bellaterra 08193, Spain.
cetto@cvc.uab.es

David W. Murray
Robotics Research Group, Department of Engineering Science
University of Oxford, Oxford OX1 3PJ, UK.
dwm@robots.ox.ac.uk

***Abstract*— In this paper we consider a single hand-held camera performing SLAM at video rate with generic 6DOF motion. The aim is to optimise both the localisation of the sensor and building of the feature map by computing the most appropriate control actions or movements. The actions belong to a discrete set (e.g. go forward, go left, go up, turn right, etc), and are chosen so as to maximise the mutual information gain between posterior states and measurements. Maximising the mutual information helps the camera avoid making ill-conditioned measurements appropriate to bearing-only SLAM. Moreover, orientation changes are determined by maximising the trace of the Fisher Information Matrix. In this way, we allow the camera to continue looking at those landmarks with large uncertainty, but from better-posed directions. Various position and gaze control strategies are first tested in a simulated environment, and then validated in a video-rate implementation. Given that our system is capable of producing motion commands for a real-time 6DOF visual SLAM, it could be used with any type of mobile platform, without the need of other sensors.**

I. Introduction

Impressive advances in 2D and, more recently, 3D simultaneous localisation and mapping (SLAM) for mobile robots have been made over the last 15 years, largely using sonar and laser range sensing [1]–[5]. Most recently, there has been considerable interest in solving the SLAM problem using visual sensing, both in order to obtain more accurate 3D representations of the environment and to exploit its richer potential for scene representation [6], [7]. In this communication, we consider the problem of SLAM with a single camera carried by a human, and how to implement control strategies in this context. In that sense, this work is different from other control work because we can only give a human quite approximate, low frequency, easy to understand commands like 'left', 'right', 'stay'.

One of the first active vision-based SLAM approaches used feature correspondences from stereo image pairs [6]. The computational burden for the accurate detection and matching of image pairs motivated the use of active visual sensing for landmark selection in sparse feature maps. Their work is different to ours because they only control orientations of the stereo head, and we are now talking about actually controlling translation as well. Other reported techniques to visual SLAM — although with no control — include the use of SIFT features, and matching over a trinocular rig [7]. More elegantly and economically, feature locations can also be computed by tracking landmarks over multiple views from only one camera, a process referred to a 'bearing-only SLAM'.

One key issue in bearing-only SLAM is the initialisation of feature locations. In [8] for example, the initial estimation of a landmark's location is achieved by sampling hypotheses of a 1D particle distribution along the line of sight. Another technique consists of using sums of Gaussian distributions to parameterise 3D feature locations over a delayed state representation [9].

When the sensor capabilities in SLAM are limited, camera motion plays an important role in the quality of reconstruction obtained. Driving the sensor to the locations that maximise the expected information gain from acquiring an observation at that location has been a common strategy [10]–[12]. However, Sim has showed that maximising the expected information gain leads to ill-conditioned filter updates in the bearing-only SLAM [13]. In [14], Bryson *et al.* present simulated results of the effect different vehicle actions have with respect to the entropic mutual information gain. The analysis is performed for a 6DOF aerial vehicle equipped with two cameras and an inertial sensor, for which landmark range, azimuth, and elevation readings are simulated, and data association is known.

In this paper we are interested in the video-rate estimation *and control* of a single camera's motion, moving rapidly with 6DOF in 3D in normal human environments, mapping visual features with minimal prior information about motion dynamics. Our aim is to localise the sensor and build a feature map by computing the appropriate control actions in order to improve overall system estimation.

However, insisting on video-rate performance using modest hardware imposes severe restrictions on the volume of

This research is supported by the Spanish Ministry of Education and Science under projects DPI 2004-5414, TIC 2003-09291, and the EU PACO-PLUS project FP6-2004-IST-4-27657 to TVC and JAC, and by the UK Engineering and Physical Research Council under an Advanced Research Fellowship Grant GR/N03266 to AJD, and by Grant GR/S97774 to DWM.

computation that can take place in each 33ms time step. Re-estimation must take place of course, but making strictly optimal camera movements would require in addition the computation of the derivatives of a well-chosen performance metric with respect to the inputs [15]. Such a computation remains unfeasible for a 6DOF highly nonlinear system model. Besides, human actions can only be approximate, and at low frequency. So, instead of computing the optimal motion command, we decide only upon a small set of choices.

Actions belong to a discrete set (eg. go forward, go left, go up, turn right, etc.), and the particular movement chosen is the one that maximises the mutual information gain between posterior states and measurements. Using entropy for exploration only makes sense if we can be certain that uncertainty is reduced as landmarks are being discovered. To that, one must have an idea first of the shape of the space to be mapped, and filling it with randomly placed features with large uncertainty [14]. Maximising the mutual information aims at reducing the overall state uncertainty, and helps the camera move away from making repeated ill-conditioned measurements. Orientation changes are determined by maximising the trace of the Fisher Information Matrix. In this way, we allow the camera still to look at those landmarks with large uncertainty, but from better-posed directions.

The remainder of the paper is ordered as follows. First we briefly describe the system and the estimation scheme. Then the metrics used as cost functions to choose the appropriate actions are explained; and our control strategy is illustrated through simulations. Lastly, we present the results of real-time experiments with a hand-held wide-angle camera, where a GUI feeds-back motion commands to the user.

II. 6 DOF Bearing-Only SLAM

A. Unconstrained Camera Motion

It is assumed that the camera could be attached to any mobile platform — in our case the hand — and is free to move in any direction in $\mathrm{I\!R}^3 \times SO(3)$. We adopt a smooth unconstrained constant-velocity motion model, its translational and rotational altered only by zero-mean, normally distributed accelerations and staying the same on average. The Gaussian acceleration assumption means that large impulsive changes of direction are unlikely. The camera motion prediction model is

$$\mathbf{x}_{v(k+1|k)} = \begin{bmatrix} \mathbf{p}_{(k+1|k)} \\ \mathbf{q}_{(k+1|k)} \\ \mathbf{v}_{(k+1|k)} \\ \boldsymbol{\omega}_{(k+1|k)} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_{(k|k)} + (\mathbf{v}_{(k|k)} + \mathbf{a}_{(k)}\Delta t)\Delta t \\ \boldsymbol{\mathcal{Q}}\mathbf{q}_{(k|k)} \\ \mathbf{v}_{(k|k)} + \mathbf{a}_{(k)}\Delta t \\ \boldsymbol{\omega}_{(k|k)} + \boldsymbol{\alpha}_{(k)}\Delta t \end{bmatrix},$$

with $\mathbf{p} = [x, y, z]^\top$ and $\mathbf{q} = [q_0, q_1, q_2, q_3]^\top$ denoting the camera pose (three states for position and four for orientation using a unit norm quaternion representation), and $\mathbf{v} = [v_x, v_y, v_z]^\top$ and $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^\top$ denoting the linear and angular velocities, respectively. The subscripts $(k|k)$ and $(k+1|k)$ denote the posterior at time k and the prior (before integrating measurements) at $k+1$. The input to the system is the acceleration vector $\mathbf{u} = [\mathbf{a}^\top, \boldsymbol{\alpha}^\top]^\top = [a_x, a_y, a_z, \alpha_x, \alpha_y, \alpha_z]^\top$.

An Extended Kalman Filter propagates the camera pose and velocity estimates, as well as feature estimates. A state that includes the features $\mathbf{y}$ is made of $\mathbf{x} = [\mathbf{x}_v^\top, \mathbf{y}^\top]^\top$. The model $\boldsymbol{\mathcal{Q}}$ for the prediction of change in orientation is inspired by [16] and is detailed in the Appendix. The redundancy in the quaternion representation is removed by a $||\mathbf{q}|| = 1$ normalisation at each update, accompanied by the corresponding Jacobian modification.

B. Feature Extraction

In this work we are interested in mapping the 3-D coordinates of salient point features from images, and need to do so at video-rate. As in previous work, we use the Shi-Tomasi saliency operator, and match correspondences in subsequent frames using normalised sum-of-squared differences [6], [8]. Although more robust detectors such as SIFT have become widely popular for their ability to find and match features with higher degree of uniqueness, they come at the expense of heavier computational load.

Image projection is modelled using a full perspective wide angle camera. The position of a 3D scene point $\mathbf{y}_i$ is transformed into the camera frame as $\mathbf{y}_i^c = [x^c, y^c, z^c]^\top = \boldsymbol{\mathcal{R}}^\top(\mathbf{y}_i - \mathbf{p})$, with $\boldsymbol{\mathcal{R}}$ the rotation matrix equivalent of $\mathbf{q}$. The point's projection onto the image plane is

$$\mathbf{h}_i = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u_0 - u_c/\sqrt{d} \\ v_0 - v_c/\sqrt{d} \end{bmatrix}, \qquad (1)$$

where $u_c = fk_u x^c/z^c$, $v_c = fk_v y^c/z^c$, the radial distortion term is $d = 1 + K_d(u_c^2 + v_c^2)$, and the intrinsic calibration of the camera — focal distance f, principal point (u_0, v_0), pixel densities k_u and k_v, and radial distortion parameter K_d — are determined beforehand.

When an image feature is detected, its measurement must either be associated with an existing feature or be added as a new feature in the map. The location of the camera, along with the locations of the already mapped features, are used to predict feature position $\mathbf{h}_i$ using Eq. (1), and these estimates checked against the measurements using a nearest neighbour test. Feature search is constrained to 3σ elliptical regions around the image estimates as defined by the innovation covariance matrix $\mathbf{S}_i = \mathbf{H}_i\mathbf{P}_{k+1|k}\mathbf{H}_i^\top + \mathbf{R}$, with $\mathbf{H}_i$ the Jacobian of the sensor model with respect to the state, $\mathbf{P}_{k+1|k}$ the prior state covariance, and measurements $\mathbf{z}_i$ assumed corrupted by zero mean Gaussian noise with covariance $\mathbf{R}$.

C. Initialisation

Inserting a new feature to the map cannot be done immediately because the measurement model is non-invertible. Though bearing is recoverable from one measurement, 3D depth is not.

Several schemes have been reported [8], [9], [17], and we adopt the first of these. The initial measurement results in a semi-infinite line with Gaussian uncertainty in its parameters, starting at the estimated camera position and heading to infinity along the feature viewing direction. A 1D particle distribution represents the likelihood of the 3D feature's position along this line. The line is projected as an epipolar line into subsequent images, but specifically it is the projection of the point particles and their uncertainly ellipses that provide the regions to be searched for a match, in turn producing likelihoods for Bayesian re-weighting of the depth distribution. A small number of steps is required to reduce to below a threshold the ratio of the standard deviation in depth to the depth estimate itself. At that time, the depth distribution is re-approximated as Gaussian and the feature is initialised as a 3D point $\mathbf{y}_i$ into the map.

III. Information Gain

This section first presents a metric for expected information gain as a result of performing a given action, and then develops an overall information conditioning strategy for the computation of orientations. The aim will be to move the camera in the direction that most reduces the uncertainty in the entire SLAM state, by using the information that should be *gained* from future, predicted, landmark observations were such a move to be made, but taking into account the information *lost* as a result of moving with uncertainty.

A. Mutual Information Gain

We adopt entropy as a measure of uncertainty; that is, as a measure of how much randomness there is in our state estimate. Entropy is defined as $H(X) = -\sum_{\mathbf{x}} p(\mathbf{x}) \log p(\mathbf{x})$, which, for our case where $p(\mathbf{x})$ is a n-variate Gaussian distribution, reduces to $H(X) = \frac{1}{2}\log((2\pi)^n|\mathbf{P}|)$.

Now consider the following two random vectors: the state prior $\mathbf{x}_{k+1|k}$, and the prediction of measurement i, $\mathbf{z}_{i,k+1|k}$. We want to choose the action that maximises the mutual information between the two. The mutual information is defined as the relative entropy between the joint distribution $p(\mathbf{x},\mathbf{z}_i)$, and the marginals $p(\mathbf{x})$ and $p(\mathbf{z}_i)$.

$$\begin{aligned} I(X;Z) &= \sum_{\mathbf{x}\in X, \mathbf{z}_i \in Z} p(\mathbf{x},\mathbf{z}_i) \log \frac{p(\mathbf{x},\mathbf{z}_i)}{p(\mathbf{x})p(\mathbf{z}_i)} \\ &= H(X) + H(Z) - H(X,Z) \\ &= H(X) - H(X|Z)\,, \end{aligned}$$

which, for our Gaussian multivariate case, evaluates to

$$\begin{aligned} I(X;Z) &= \frac{1}{2}\log\left(\frac{|\mathbf{P}_{\mathbf{x}}|}{|\mathbf{P}_{\mathbf{x}} - \mathbf{P}_{\mathbf{xz}}\mathbf{P}_{\mathbf{z}}^{-1}\mathbf{P}_{\mathbf{xz}}^{\top}|}\right) \\ &= \frac{1}{2}\log\left(\frac{|\mathbf{P}_{k+1|k}|}{|\mathbf{P}_{k+1|k} - \mathbf{P}_{k+1|k}\mathbf{H}_i^{\top}\mathbf{S}_i^{-1}\mathbf{H}_i\mathbf{P}_{k+1|k}^{\top}|}\right) \\ &= \frac{1}{2}\left(\log|\mathbf{P}_{k+1|k}| - \log|\mathbf{P}_{k+1|k+1}|\right)\,. \end{aligned}$$

Thus, in choosing a maximally mutually informative motion command, we are maximising the difference between prior and posterior entropies [18]. In other words, we are choosing the motion command that most reduces the uncertainty of $\mathbf{x}$ due to the knowledge of $\mathbf{z}$ as a result of a particular action. Figure 1 shows the directions maximising the mutual information for a simple 2DOF camera and 3 landmarks.

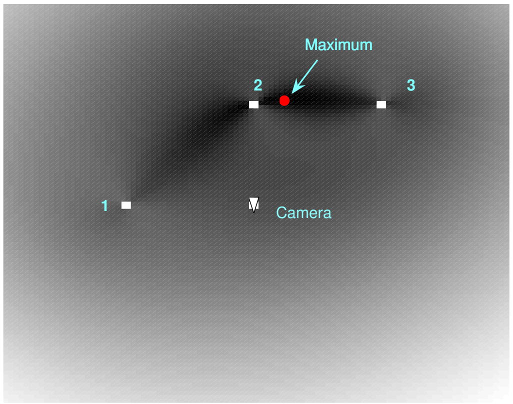


Fig. 1. Maximisation of mutual information for the evaluation of motion commands. A simple 2DOF camera is located at the centre of the plot, and a decision where to move must be taken as a function of the pose and landmark states, and the expected measurements. Three landmarks are located to its left, front, and right-front. Moving to the location in between landmarks 2 and 3 maximises the mutual information between the SLAM prior and the measurements for this particular example.

Note that the use of mutual information only makes sense prior to reaching full correlation. In SLAM, $|\mathbf{P}_{k|k}|$ tends asymptotically to zero, point at which the map becomes fully correlated and there is nothing else the camera can do to improve the estimates of the features. From then on, entropy can still be used to decide what actions to take to reduce the camera's own uncertainty, and this can be done just by replacing $\mathbf{x}$ with $\mathbf{x}_v$ from the above discussion.

B. Fisher Information for Gaze Direction

Measurements in the bearing-only SLAM case are ill-posed for motions along the principal axis, when points are close to the principal axis and there is little perspective distortion. Motion commands based on the maximisation of the mutual information metric drive the camera away from those configurations, that is, perpendicular to the principal axis. However, we still want the camera to look at those landmarks with large uncertainty so as to reduce their covariance when seen from different locations. To do that, we incorporate another information metric to control the direction of gaze. From a set of possible orientation changes, we propose choosing that which maximises the trace of the Fisher Information Matrix. In this way we will be choosing the best direction to look at, in the sense that it is the one that is most informative, but from a different position than the ill-posed one. Under the Gaussian assumption for sensor and platform noises, the minimisation of the least squares criteria (the KF) is equivalent to the maximisation of a likelihood function $\Lambda(\mathbf{x})$ given the set of observations Z^k, that is, the maximisation of the joint pdf of the entire history of observations, $\Lambda(\mathbf{x}) = \prod_{i=1}^{k} p(\mathbf{z}_i|\mathbf{x}, Z^{i-1})$.

The Total Fisher Information Matrix, a quantification of the maximum existing information in the observations about the state, is defined in [19] as the expectation $\mathbf{J} = E\left[\left(\nabla \log \Lambda\left(\mathbf{x}\right)\right)\left(\nabla \log \Lambda\left(\mathbf{x}\right)\right)^{\top}\right]$, which here evaluates to $\mathbf{J} = \sum \mathbf{H}^{\top}\mathbf{S}^{-1}\mathbf{H}$.

The information for the reconstruction of the state contributed by the set of measurements at each iteration is contained in $\mathbf{H}^{\top}\mathbf{S}^{-1}\mathbf{H}$. The eigenvalues λ_j of this contribution to $\mathbf{J}$ show which linear combinations of the states can be estimated with good accuracy and which will have large uncertainties from the coming measurements. It also shows which linear combinations of states are unobservable. When one dimension of $\mathbf{J}$ has a very small eigenvalue (information along the line of sight), the product is not a reliable measure of the elongation of the information hyperellipsoid, as it collapses the volume to zero. Our strategy is to look in the direction at which $\sum \lambda_j$ is maximum [20]. This is the viewing direction that will introduce the largest amount of information in one single measurement step.

Under a Fisher information motion strategy, maximally informative actions move the robot as close as possible to the landmarks under observation. We do not want to move towards them, but only to orient towards them. Our idea of using the Fisher Information is only to fixate our camera to those most uncertain landmarks, and use the change in entropy to select movement actions. This way, by using the mutual information metric, maximally informative actions would prevent the camera from producing ill-posed measurements. Note that an omnidirectional sensor would not require a strategy to direct fixation. In our case, as opposed to a mobile robot, translation and orientation changes are kinematically decoupled, for this reason, it makes sense to use different information measures in evaluating them.

IV. Control Strategy

In this Section we demonstrate in simulation how combining the strategies of effectively controlling translation by maximising mutual information thereafter controlling orientation by maximising the information available from the new position yields reliable active control of pose and velocity for a free moving camera, whilst building a map optimally.

A. Deciding Where to Go and Where to Look At

As noted earlier, the real-time requirements of the task preclude using an optimal control decision that takes into account all possible motion commands which is impracticable to compute, leading to an exponential growth because of the curse of dimensionality of long term action evaluation. Instead we evaluate our information metrics for a small set of actions carried out over a fixed amount of time, and choose the best action from those.

The set of possible actions is divided in two groups. Mutual information is evaluated for the translational actions `go_forward`, `go_backwards`, `go_right`, `go_left`, `go_up`, `go_down`, and `stay`; and Fisher information is maximised from the set of orientation commands `turn_right`, `turn_left`, and `stay`.

In our simulated setting, desired camera locations are predicted for the best action chosen, and a PD low-level control law is applied to ensure these locations are reached at the end of one second; at which point the motion metric is again evaluated to determine the next desired action. Orientations however, are evaluated at frame rate, leaving the system to freely rotate, governed only by the information maximisation strategy.

The simulation considers a fixed number of expected landmarks to be found, and both the Mutual Information and Fisher Information metrics are computed taking into account the corresponding full covariance matrices, including these unvisited landmarks, which have been initialised with large uncertainties. This is the only thing that prevents our control strategies from defaulting to homeostasis.

B. Simulation Results

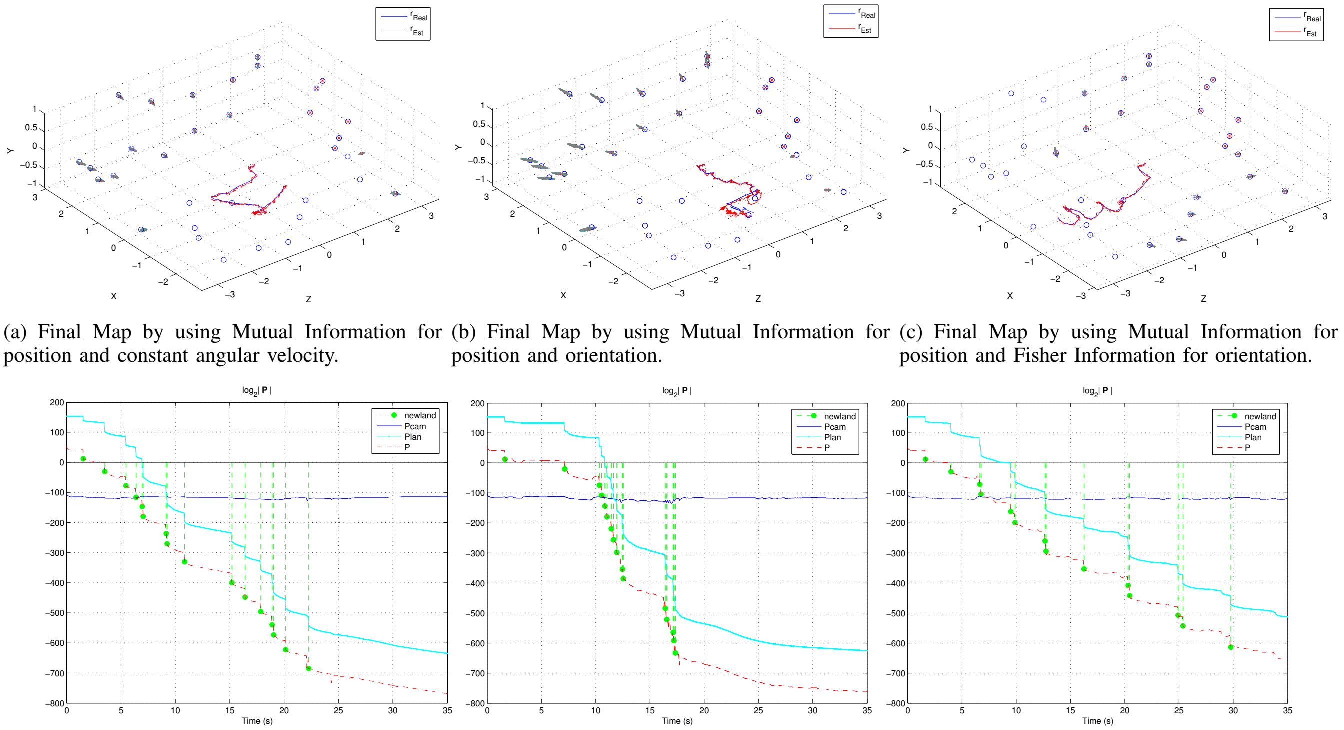
Figure 2 contains simulation results from our mutual information strategy for the computation of motion commands, and compares various orientation computation schemes. The simulated environment represents a room $6 \times 6 \times 2$ m^3 in size containing 33 randomly distributed point landmarks, out of which 6 are fiduciary points, to be used as global references [21].

The initial standard deviation in camera pose is 6-cm in the x and y directions, 4.6 cm in height z, and 45° in orientation, right after matching the fiduciary points, but before any motion takes place. Sensor standard deviation is set at 2 pixels, and data association is not known a priori. Instead, nearest neighbour χ-squared tests are computed to guarantee correct matching. New features are initialised once their ratio of depth estimate to depth standard deviation falls below a threshold of 0.3.

The plots show the results of actively moving a 6-DOF camera whilst building a map of 3D features. In all cases, each of the seven motion actions will produce a displacement of 30 cm in the corresponding direction. Our mutual information metric is evaluated at each of these positions. The action that maximises the metric is chosen, and the camera is controlled to reach that position in one second with a PD control law. Orientation changes are computed every 50 ms.

Three approaches were tested for the computation of gaze commands: (i) constant rotational velocity of 0.2 rad/sec, frames (a,d); (ii) maximisation of mutual information both for the position and orientation of the moving camera, frames (b,e); and (iii) maximisation of mutual information for position and maximisation of Fisher information for gaze, frames (c,f). The experiment shown in the plots lasted 35 seconds.

The constant rotational velocity and the mutual information strategies tend to insert landmarks into the map at a faster pace than the Fisher Information strategy. As can be seen in the error plots in Figure 3, this might not be always the best choice. It seems reasonable to let the system accurately locate the already seen landmarks before actively searching for new ones.



(a) Final Map by using Mutual Information for position and constant angular velocity.

(b) Final Map by using Mutual Information for position and orientation.

(c) Final Map by using Mutual Information for position and Fisher Information for orientation.

(d) Entropy for MI in position and constant angular velocity.

(e) Entropy for MI in position and orientation.

(f) Entropy for MI in position and FI in orientation.

Fig. 2. Trajectories with Final Maps and Entropy. ($\mathtt{r}_{\mathtt{Real}}$ and $\mathtt{r}_{\mathtt{Est}}$ are the real and estimated camera trajectories, the label newland and the green dots and dotted vertical lines represent the value of entropy at the instant when new landmarks are initialised. Pcam, Plan, and P indicate the camera, map, and overall entropies.

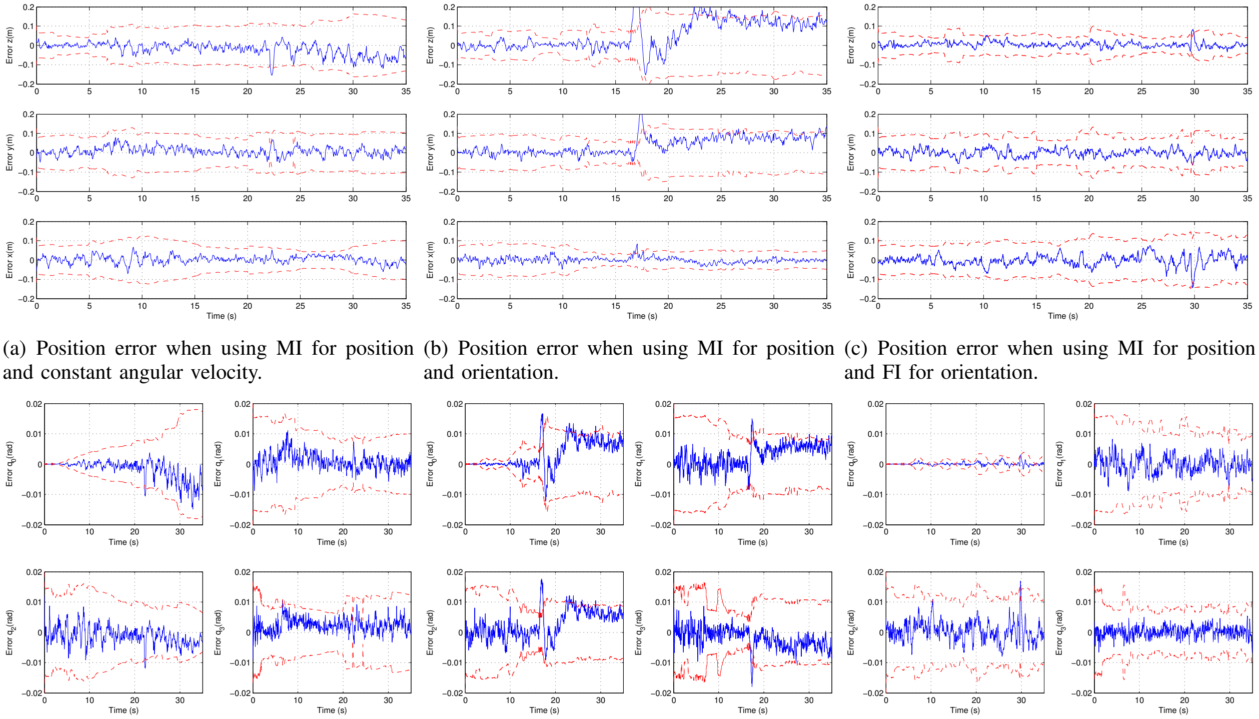
The third alternative, controlling camera orientation by maximising the Fisher Information entering into the filter, has the effect that it focuses on reducing the uncertainty of the already seen landmarks, instead of eagerly exploring the entire room for new landmarks. The reason is that landmarks that have been observed for a small period of time still have large depth uncertainty, and the Fisher Information metric is maximised when observations are directed towards them. The technique tends to close loops at a faster pace than the other two approaches, thus propagating correlations amongst landmarks and poses in a more efficient way. Additionally, by revisiting fiduciary points more often, orientations are much better estimated in this case.

Strategy (iii) needs more time to reduce entropy and takes more time to insert the same number of landmarks in the map. But, at the point at which the same number of landmarks is available it has lower entropy than the other two strategies (see for example in Figure 2, frames (d-f), that when the 14th landmark is added, the times are 19, 18, and 30 secs, and the entropies are -530, -550, and -610).

V. Experiments

This section presents an initial experimental result validating the maximisation of mutual information strategy for the control of a hand-held camera in a challenging 15fps visual SLAM application. Within a room, the camera starts approximately at rest with some known object in view to act as a starting point and provide a metric scale to the proceedings. The camera moves, translating and rotating freely in 3D, according to the instruction provided in a graphical user interface, and executed by the user, within a room or a restricted volume, such that various parts of the unknown environment come into view. The aim is to estimate and control the full camera pose continuously during arbitrarily long periods of movement. This involves accurately mapping (estimating the locations of) a sparse set of features in the environment.

Given that the control loop is being closed by the human operator, only displacement commands are computed. Gaze control is left to the user. Furthermore, the mutual information measure requires evaluating the determinant of the full covariance matrix at each iteration. Because of the complexity of this operation, single motion predictions are evaluated one frame at a time. It is only until the 15th frame in the sequence that all mutual information measures are compared, and a desired action is displayed on screen. That is, the user is presented with motion directions to obey every second. Note also, that in computing the mutual information measure, only the camera position and map parts of the covariance matrix are used, leaving out the gaze and velocity parts of the matrix. Finally, to keep it running in real-time, the resulting application must be designed for sparse mapping. That is, with the computing capabilities of an off-the-shelf system, our current application



(a) Position error when using MI for position and constant angular velocity.
(b) Position error when using MI for position and orientation.
(c) Position error when using MI for position and FI for orientation.

(d) Orientation error when using MI for position and constant angular velocity.
(e) Orientation error when using MI for position and orientation.
(f) Orientation error when using MI for position and FI for orientation.

Fig. 3. Estimation errors for camera position and orientation and their corresponding 2σ variance bounds. Position errors are plotted as x, y, and z distances to the real camera location in meters, and orientation errors are plotted as quaternions.

is limited to less than 50 landmarks.

Figure 4 shows the graphical user interface. The top part of the figure contains a 3D plot of the camera and the landmarks mapped, while the bottom part shows the information being displayed to the user superimposed on the camera view. Figure 5 contains a plot of the decrease in the various entropies for the map being built, and the list of actions chosen as shown to the user during the first minute.

Worth noticing is that in the real-time implementation, the system prompts the user for repeated up-down movements, as well as left-right commands. This can be explained as if after initialising new features, the system repeatedly asks for motions perpendicular to the line of sight to best reduce their uncertainty. Also, closing loops has an interesting effect in the reduction of entropy, as can be seen around the 1500th frame on Fig. 5-a.

VI. Conclusion

In conclusion, we have shown plausible motion strategies in a video-rate visual SLAM application. On the one hand, by choosing a maximal mutually informative motion command, we are maximising the difference between prior and posterior SLAM entropies, resulting in the motion command that mostly reduces the uncertainty of $\mathbf{x}$ due to the knowledge of $\mathbf{z}$. Alternatively, by controlling gaze maximising the information about the measurements, we get a system that prioritises in accurately locating the already seen landmarks before actively searching for new ones.

Our method is validated in a video-rate hand-held visual SLAM implementation. Given that our system is capable of producing motion commands for a real-time 6DOF visual SLAM, it is sufficiently general to be incorporated into any type of mobile platform, without the need of other sensors.

A possible weakness of this information-based approach is that it estimates the utility of measurements assuming that our models are correct. Model discrepancies, and effects of linearisation in the computation of our estimation and control commands might lead to undesirable results.

Appendix

The orientation of the camera frame, and its rate of change, are related to the angular velocity by the quaternion multiplication $\boldsymbol{\Omega} = 2\dot{\mathbf{q}}\mathbf{q}^*$, with $\boldsymbol{\Omega} = [0, \omega_x, \omega_y, \omega_z]^\top$, the angular velocity vector expressed in quaternion form, and $\mathbf{q}^*$ is the orientation quaternion conjugate. Or equivalently, by $\dot{\mathbf{q}} = \frac{1}{2}\mathbf{M}\mathbf{q} \approx \frac{\mathbf{q}_{(k+1)} - \mathbf{q}_{(k)}}{\Delta t}$, with

$$\mathbf{M} = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & -\omega_z & \omega_y \\ \omega_y & \omega_z & 0 & -\omega_x \\ \omega_z & -\omega_y & \omega_x & 0 \end{bmatrix} .$$

Solving for $\mathbf{q}_{(k+1)}$ in the above approximation when $\boldsymbol{\omega}$ is constant, our smooth motion model for the prediction of change

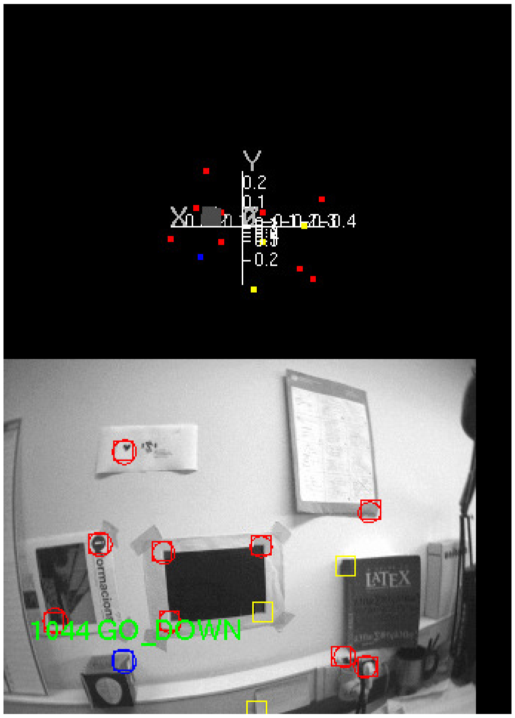
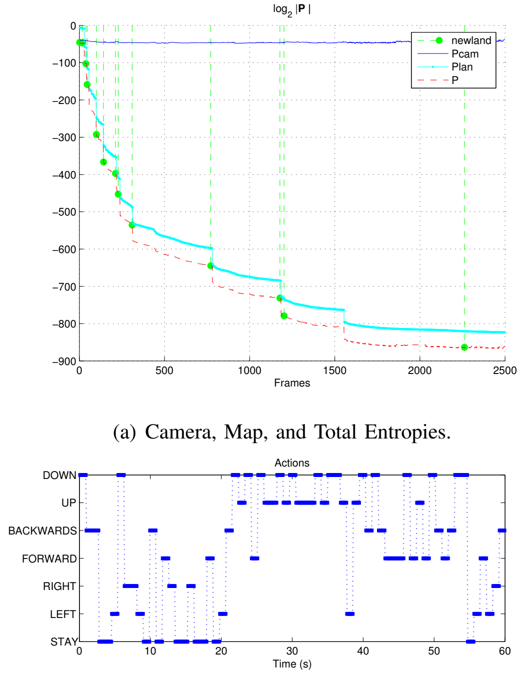


Fig. 4. Feature map and camera view as shown in the Graphical User Interface (844th frame).



(a) Camera, Map, and Total Entropies.

(b) Actions for the first minute

Fig. 5. Real-time Active Vision SLAM.

in orientation becomes $\mathbf{q}_{k+1} = \mathcal{Q}\mathbf{q}_k$ with the quaternion transition matrix

$$\mathcal{Q} = \cos\left(\frac{\Delta t\|\mathbf{\Omega}\|}{2}\right)\mathbf{I} + \frac{2}{\|\mathbf{\Omega}\|}\sin\left(\frac{\Delta t\|\mathbf{\Omega}\|}{2}\right)\mathbf{M}\,.$$

Note that when computing the quaternion propagation, the angular velocities are to be evaluated at $(k+1|k)$, i.e., including the angular acceleration term.

REFERENCES

[1] R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *Int. J. Robot. Res.*, vol. 5, no. 4, pp. 56–68, 1986.

[2] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Trans. Robot. Automat.*, vol. 17, no. 3, pp. 229–241, Jun. 2001.

[3] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Auton. Robot.*, vol. 4, no. 4, pp. 333–349, 1997.

[4] U. Frese, P. Larsson, and T. Duckett, "A multigrid algorithm for simultaneous localization and mapping," *IEEE Trans. Robot.*, vol. 21, no. 2, pp. 1–12, 2005.

[5] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous localization and mapping with sparse extended information filters," *Int. J. Robot. Res.*, vol. 23, no. 7-8, pp. 693–716, Jul. 2004.

[6] A. J. Davison and D. W. Murray, "Simultaneous localisation and map-building uisng active vision," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 7, pp. 865–880, Jul. 2002.

[7] S. Se, D. Lowe, and J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," *Int. J. Robot. Res.*, vol. 21, no. 8, pp. 735–758, Aug. 2002.

[8] A. Davison, W. Mayol, and D. Murray, "Real-time localisation and mapping with wearable active vision," in *Proc. IEEE Int. Sym. Mixed and Augmented Reality*, Tokyo, Oct. 2003.

[9] J. Sola, A. Monin, M. Devy, and T. Lemaire, "Undelayed initialization in bearing only SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Edmonton, Aug. 2005.

[10] P. Whaite and F. Ferrie, "Autonomous exploration: Driven by uncertainty," in *Proc. 9th IEEE Conf. Comput. Vision Pattern Recog.*, Seattle, Jun. 1994, pp. 339–346.

[11] H. Feder, J. Leonard, and C. Smith, "Adaptive mobile robot navigation and mapping," *Int. J. Robot. Res.*, vol. 18, pp. 650–668, 1999.

[12] F. Bourgault, A. Makarenko, S. Williams, and B. Grocholsky, "Information based adaptative robotic exploration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Lausanne, Oct. 2002.

[13] R. Sim, "Stable exploration for bearings-only SLAM," in *Proc. IEEE Int. Conf. Robot. Automat.*, Barcelona, Apr. 2005, pp. 2422–2427.

[14] M. Bryson and S. Sukkarieh, "An information-theoretic approach to autonomous navigation and guidance of an uninhabited aerial vehicle in unknown environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Edmonton, Aug. 2005.

[15] A. Adam, E. Rivlin, and I. Shimshoni, "Computing the sensory uncertainty field of a vision-based localization sensor," *IEEE Trans. Robot. Automat.*, vol. 17, no. 3, pp. 258–267, Jun. 2001.

[16] T. Broida, S. Chandrashekhar, and R. Chellappa, "Recursive 3-d motion estimation from a monocular image sequence," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 26, no. 4, pp. 639–656, Jul. 1990.

[17] T. Bailey, "Constrained initialisation for bearing-only SLAM," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 2, Taipei, Sep. 2003, pp. 1966–1971.

[18] D. J. C. MacKay, "Information based objective functions for active data selection," *Neural Comput.*, vol. 4, no. 4, pp. 589–603, 1992.

[19] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York: John Wiley & Sons, 2001.

[20] R. Sim and N. Roy, "Global A-optimal robot exploration in SLAM," in *Proc. IEEE Int. Conf. Robot. Automat.*, Barcelona, Apr. 2005, pp. 673–678.

[21] J. Andrade-Cetto and A. Sanfeliu, "The effects of partial observability when building fully correlated maps," *IEEE Trans. Robot.*, vol. 21, no. 4, pp. 771–777, Aug. 2005.

2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Depth from the visual motion of a planar target induced by zooming

Guillem Alenyà, Maria Alberich and Carme Torras

Abstract— **Robot egomotion can be estimated from an acquired video stream up to the scale of the scene. To remove this uncertainty (and obtain true egomotion), a distance within the scene needs to be known. If no a priori knowledge on the scene is assumed, the usual solution is to derive "in some way" the initial distance from the camera to a target object. This paper proposes a new, very simple way to obtain such a distance, when a zooming camera is available and there is a planar target in the scene. Similarly to "two-grid calibration" algorithms, no estimation of the camera parameters is required, and no assumption on the optical axis stability between the different focal lengths is needed. Quite the reverse, the non stability of the optical axis between the different focal lengths is the key ingredient that enables to derive our depth estimate, by applying a result in projective geometry. Experiments carried out on a mobile robot platform show the promise of the approach.**

I. INTRODUCTION

This paper presents a new method for inferring depth information using a zooming camera. In previous works [1], [2] we have shown how to recover robot egomotion from the deformation of an active contour. We have proposed to express the deformation of the contour in the image with a 6-dimensional affine *shape vector*. Then, with a non-linear non-derivable algorithmic function the performed 3D motion can be recovered up to a scale factor (as it is common in monocular vision). Scaled 3D motion can be recovered also in the context of a zooming camera [3]. Studying further the characteristics of the proposed affine shape space, we will show how the initial distance can be computed from the affine shape deformation caused by a zoom-lens camera.

Being based on active contour tracking, our egomotion recovery algorithm requires that the whole object projection keeps into the image all along the robot trajectory. This is sometimes too restrictive with a fixed camera, as the allowed robot motion is highly limited. One of the more promising solutions we have considered is to provide motion to the camera by means of a pan-and-tilt unit, and to implement a control algorithm to keep the target centered in the image (or at least within the image) in the whole sequence. One of the main problems of the control algorithm is that different gains should be applied depending on the distance from camera to target. Observe that, as usual in monocular imaging, it is not possible to disambiguate a priori the motion of a closer and small object from that of a far and big one.

Metric egomotion may be obtained if some additional information can be gathered. The scale factor depends on the camera focal distance and also on the initial distance from the camera to the viewed target. The camera focal distance can be obtained easily by a camera calibration or autocalibration method, even for zooming cameras [4]. The initial distance from camera to target is harder to obtain. In [2] we used a laser and other authors have proposed, for example, to use the range scanner of an autofocus camera [5], stereo correspondence, trifocal tensors [6], depth from defocused images [7] and depth from zooming.

In *depth from zooming* both camera and scene should be stationary and image deformation be caused only by zooming. Ma and Olsen [8] proposed a method to recover depth information from the variation in the focal distance and the optical flow. They noticed that the equation that describes the displacement obtained by zooming is similar to the one describing the translation of a camera along the optical axis. They assumed a thin-lens camera model (that nowadays is known not to be the most suitable model for zoom lenses [6]). In their mathematical formulation, they assumed that the apparent object translation is due exclusively to focal length variation. Lavest et al. [9] showed that this is not correct. In their work they use the thick-lens camera model, which is more accurate in modelling the focal change process. The correspondence that they establish between a thick-lens model and the corresponding pinhole configuration is interesting. To obtain good reconstruction data, a very accurate calibration process should be performed, including intrinsic (with radial distorsion) and extrinsic parameters. They were forced to use high-quality lenses, as they assumed that the optical axis was stable during the zooming sequence.

Rodin and Ayache [10] introduced a calibration method that does not require a physical axial camera. They used a geometric rectification method, but distorsions were not taken into account and the triangulation base they used was very small (only 50 mm).

Later, Lavest et al. [11] proposed an implicit reconstruction method that uses a two-plane geometric calibration procedure. The method was originally developed by Martins et al. [12] to solve the back-projection problem, and extended by Gremban et al. [13] to include also a solution to the projection problem, formulated with systems of linear equations. The idea is to find, without any explicit camera model, the ray in space that defines the line of sight of a given pixel. To calibrate, Lavest et al. used a micrometric table to translate the calibration pattern, as the reconstruction method that they

This work is partially funded by the EU PACO-PLUS project FP6-2004-IST-4-27657.

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

Maria Alberich is also with the Departament de Matemàtica Aplicada I, UPC, Avda. Diagonal 647, 08028 Barcelona maria.alberich@upc.edu

1-4244-0602-1/07/$20.00 ©2007 IEEE.

proposed requires a high-precision calibration process. A new point in the image (located manually in [11] and by means of an iterative algorithm in [14]) can be triangulated with the calibration data to find the 3D point location. This method has the advantages of taking into account all distorsions, the optical center displacement produced when zooming, and not requiring the estimation of the camera parameters. A common comment [15], [16] is that it doesn't take into account the blurring effects that in some situations are produced when zooming.

The article is structured as follows. Section II presents the shape space that parameterises the general 6 d.o.f motion, and the reduced space corresponding to a zooming camera used to extract the required scale. In Section III we present the calibration algorithm and the proposed method to infer depth. Experiments with real images taken from a mobile robot are explained in Section IV. Finally, in Section V some conclusions and ideas about the applicability of the method in current approaches that require an initial depth estimate are stated.

II. AFFINITY RECOVERY FROM THE DEFORMATION OF AN ACTIVE CONTOUR

Under weak-perspective conditions (i.e., when the depth variation of the viewed object is small compared to its distance to the camera), every 3D motion of a planar object projects as an affine deformation in the image plane.

The affinity relating two views is usually computed from a set of point matches [17], [18]. In this work an active contour [19] fitted to a target object is used instead. The contour, coded as a B-Spline [20], deforms between views leading to changes in the location of the control points. A relation can be established between some extracted point features and a contour, considering the list of points as the set of the B-Spline control points. As a consequence, the method presented next, that obtains a motion parameterisation through pseudoinverse multiplication, can be applied also with point correspondences (as will be proved in Sec. IV).

It has been formerly demonstrated [19], [1], [3] 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

$$\mathbf{Q}' - \mathbf{Q} = \mathbf{W}\mathbf{S} \tag{1}$$

where

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

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 $\mathbf{d}'(u)$ and $\mathbf{d}(u)$ of the planar contour:

$$\mathbf{d}'(u) = \mathbf{M}\mathbf{d}(u) + \mathbf{t}, \tag{4}$$

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 deformation of the contour parameterized as a planar affinity permits deriving the camera motion in 3D space [1] even in the presence of zooming [3]. It has shown before that different deformation spaces can be defined corresponding to several constrained robot motions [21]. I.e. in the case of a planar robot, with 3 degrees of freedom, the motion space is parameterised with two translations (T_x, T_z) and one rotation (θ_y) yielding a three-dimensional shape space, which should be enlarged with one additional degree of freedom to cope with misalignments of the camera and robot coordinate systems [2].

Here the proposed solution is similar to the one in [2]. We need to define a reduced shape space able to deal with all the possible image deformations caused by zooming. First, the effect of zooming by a factor ρ is to translate the image point x along a line going from the principal point v_0 to the point $x' = \rho x + (1 - \rho)v_0$. At practical effects, this can be implemented by multiplying the calibration matrix corresponding to the first frame by the factor ρ, and it can be introduced directly as one of the degrees of freedom in the reduced shape space that we want to build. Second, the optical axis in a zooming camera is not constant [9], since the principal point position changes when zooming. To be able to model the translation effects present when zooming, we use the horizontal and vertical translation degrees of freedom[1]. The resulting shape matrix is of the form

$$\mathbf{W_{zoom}} = \left(\begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix}, \begin{bmatrix} \mathbf{Q^x} \\ \mathbf{Q^y} \end{bmatrix} \right) \tag{5}$$

and the shape vector is

$$\mathbf{S} = [t_x, t_y, \rho] \ . \tag{6}$$

III. DEPTH FROM THE AFFINITY

As we will show, the algorithm presented here shares the main advantages of the ”two-grid calibration” algorithm [12], [11]: no estimation of the camera parameters is required, and no assumption on the optical axis stability between the different focal lengths is needed. Quite the reverse, the non stability of the optical axis between the different focal lengths is the key ingredient that enables to derive our depth estimate. Note that if we try to model a zooming camera with the pinhole model we can assume neither that the optical axis is constant nor that the projection center is at the same place [4]. We only assume that the optical axis varies always in the same way between some two given focal lengths. We also suppose that the relation between two views of the same scene taken by a static zooming camera is accurately approximated by a planar homothetic transformation (a change in scale and a translation). As explained before, the scale factor (equivalently, the ratio of the homothetic transformation) accounts for the change in focal length, and the translation accounts for the displacement of the principal point, due to the non stability of the optical axis.

[1] This can be derived in a similar manner as was done in [21].

Furthermore, the proposed algorithm overcomes one of the major difficulties of the existing algorithms: it works well under affine viewing conditions. Moreover, from a computational point of view, it is a straightforward calibration algorithm: it avoids time-consuming minimization calculations, since the input data are ratios of three planar homothetic transformations. The estimation of these ratios relies on the restriction of a planar affine shape-space, which parameterizes the deformation of the projected target in the image (see Sec. II), combined with a quick an robust feature location method, such as an active contour tracking [19] or an affine-transfer based method [22].

A. Calibration algorithm

A planar target is located at a distance z_1 of the camera. The target is viewed by the camera at zoom A. Then the camera switches to zoom B and the homothetic transformation h_1 (whose ratio will be denoted ρ_1) that relates these two views (from zoom A to zoom B) is computed. This process is repeated at a distance z_2 of the camera: a planar target (it may be different from the preceding one) is viewed by the zooming camera, from zoom A to zoom B, and the homothetic transformation h_2 (whose ratio will be denoted ρ_2) that relates the initial and final views is computed.

If a new planar target (at an unknown distance z) is acquired with the zooming camera, again from zoom A to zoom B, then the homothetic transformation h (whose ratio will be denoted by ρ) that relates the initial and final views is computed. We claim that the ratio of depths $\frac{z_2-z_1}{z-z_1}$ may be computed from the ratios of the preceding homothetic transformations and is given by $\frac{\rho(\rho_2-\rho_1)}{\rho_2(\rho-\rho_1)}$. Thus, we obtain a straightforward estimation of the unknown depth z, without knowing any camera parameter. Moreover, the tedious use of metric instruments, such as a micrometric table, is avoided in the calibration process, since the relative orientation between the planes containing the two calibration targets is not relevant; besides, there is no need to use grids, hence the two calibration targets may be familiar objects in the scene (such as a door, window, board ...). The problem of computing accurately the ratio of the homothetic transformation relating the initial and final views of a zooming camera is overcome by reducing the dimension of the shape vector, which encodes the affine relation between the two views (see Section II).

B. Inferring the depth

We will show, as announced, how the non stability of the optical axis between the different focal lengths is used to infer our depth estimate.

We suppose that the direction of the optical axis in focal length A differs slightly from the direction of the optical axis in focal length B. Hence there exists an optical ray l in zoom A, which goes through an image point x, whose direction equals the direction of the optical axis a_B in zoom B (see Fig. 1).

This ray l is close to the optical axis in zoom A, and it cuts the calibration planes in the points X_1 and X_2, and the target plane in the point X. Thus the simple ratio of these points $(X_1, X_2, X) = \frac{d(X_1,X_2)}{d(X_1,X)}$ (where $d(Y_1, Y_2)$ is the distance between two points Y_1 and Y_2) is a sharp estimate of the ratio of depths $\frac{z_2-z_1}{z-z_1}$.

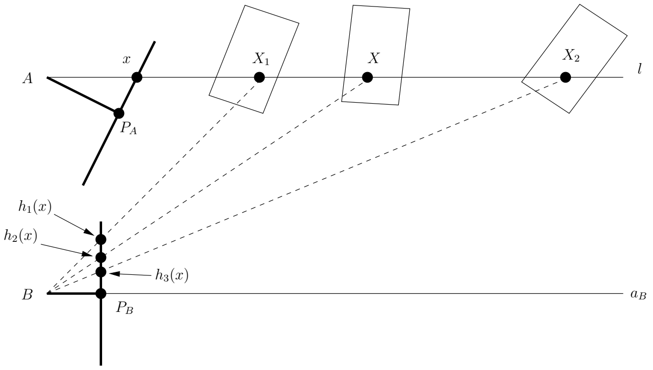


Fig. 1. A static zooming camera views the same scene with zoom A and zoom B. The variation of the optical axis between the two focal lengths has been magnified in order to exhibit the relevant features (see III-B) to infer the depth in the algorithm of Section III-A.

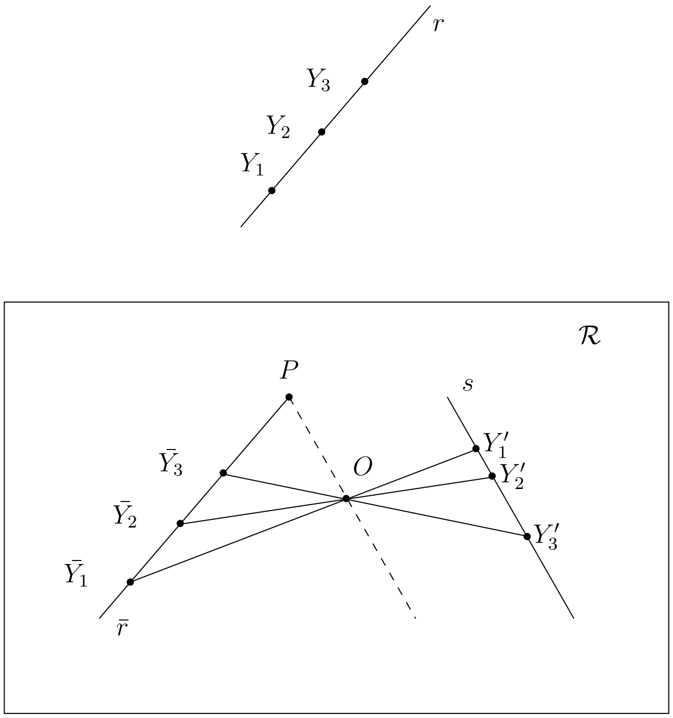


Fig. 2. Scene line r with three reference points Y_1, Y_2, Y_3 projected in the image $\mathcal{R}$ to $\bar{x}$ and $\bar{Y_1}$, $\bar{Y_2}$, $\bar{Y_3}$ respectively. P is the vanishing point of r. Auxiliary points are drawn on $\mathcal{R}$ and lines to derive the equality of simple ratios (Y_1, Y_2, Y_3) = (Y_1', Y_2', Y_3') claimed in Theorem 1.

The scene points X_1, X_2 and X are projected in zoom B to the image points $h_1(x)$, $h_2(x)$ and $h(x)$, respectively (see Fig. 1). Our goal is to determine the simple ratio of the scene points (X_1, X_2, X) from the image points $h_1(x)$, $h_2(x)$ and $h(x)$. This is done by applying the following result of projective geometry:

Theorem 1: Given the vanishing point P of a scene line r, with three reference points Y_1, Y_2, Y_3, then the simple

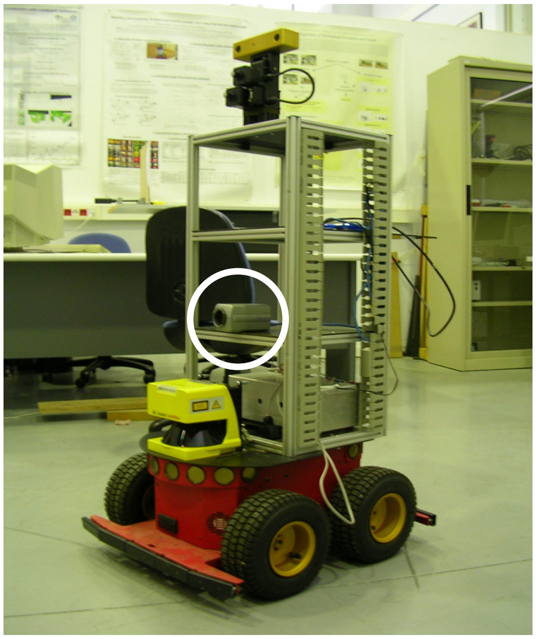


Fig. 3. Pioneer 3AT mobile platform used in the experiments

ratio (Y_1, Y_2, Y_3) can be computed from their imaged points $\overline{Y}_1$, $\overline{Y}_2$, $\overline{Y}_3$ as follows: choose an image point O (not on the imaged line $\overline{r}$) and an image line s (not going through O) parallel to the line joining O and P; for $i = 1, 2, 3$, determine the point Y_i' lying on s and on the line joining O and $\overline{Y}_i$; then $(Y_1, Y_2, Y_3) = (Y_1', Y_2', Y_3')$ (see Fig. 2).

The case that concerns us is when $r = l$ and $Y_1 = X_1$, $Y_2 = X_2$, $Y_3 = X$. The vanishing point of l (the image point of the point at infinity of l) is the principal point $P_B = P$ in zoom B. The assumption that the optical axis varies always in the same way between zoom A to zoom B is equivalent to $P_B = h_1(P_A) = h_2(P_A) = h(P_A)$, where P_A is the principal point in zoom A. Therefore, if we fix an image reference system centered at $P = P_B$, with first vector in the direction of $\overline{r}$ and unit length $d(x, P_A)$, then $h_1(x)$, $h_2(x)$ and $h(x)$ have coordinates $(\rho_1, 0)$, $(\rho_2, 0)$ and $(\rho, 0)$, respectively. By choosing, for instance, $O = (0, -1)$ and the line $x = 1$, and by applying Theorem 1, we obtain the desired result

$$(X_1, X_2, X) = \frac{\rho(\rho_2 - \rho_1)}{\rho_2(\rho - \rho_1)}. \qquad (7)$$

IV. EXPERIMENTS

The performance of the proposed algorithm has been tested on real images acquired with a Sony DFW-VL500 digital camera. The camera brochure states that the zoom of the camera can be moved to predefined positions ranging from 40 to 1432 corresponding to focal lengths from 5.5 to 64 mm. The camera is mounted on a Pioneer mobile platform (see Fig. 3). The translations performed with the robot are roughly estimated with marks on the floor. The drawers of a table and a stool serve as *natural* landmarks from which calibration information is extracted. Although the focus of the camera is kept constant, no defocus problems have been observed in the range of zoom positions and distances that we have used.

TABLE I
RESULTS OF ESTIMATED DEPTHS USING DIFFERENT CALIBRATION DISTANCES AND DIFFERENT TARGET OBJECTS.

Exp. ID	Cal1	Cal2	Estimated	Measurements
1	240	360	277.6	280
2			321.4	320
3			401.7	400
4			269.8	280
5	240	320	288.2	280
6			357.8	360
7			281.6	280
8	320	360	367.7	400

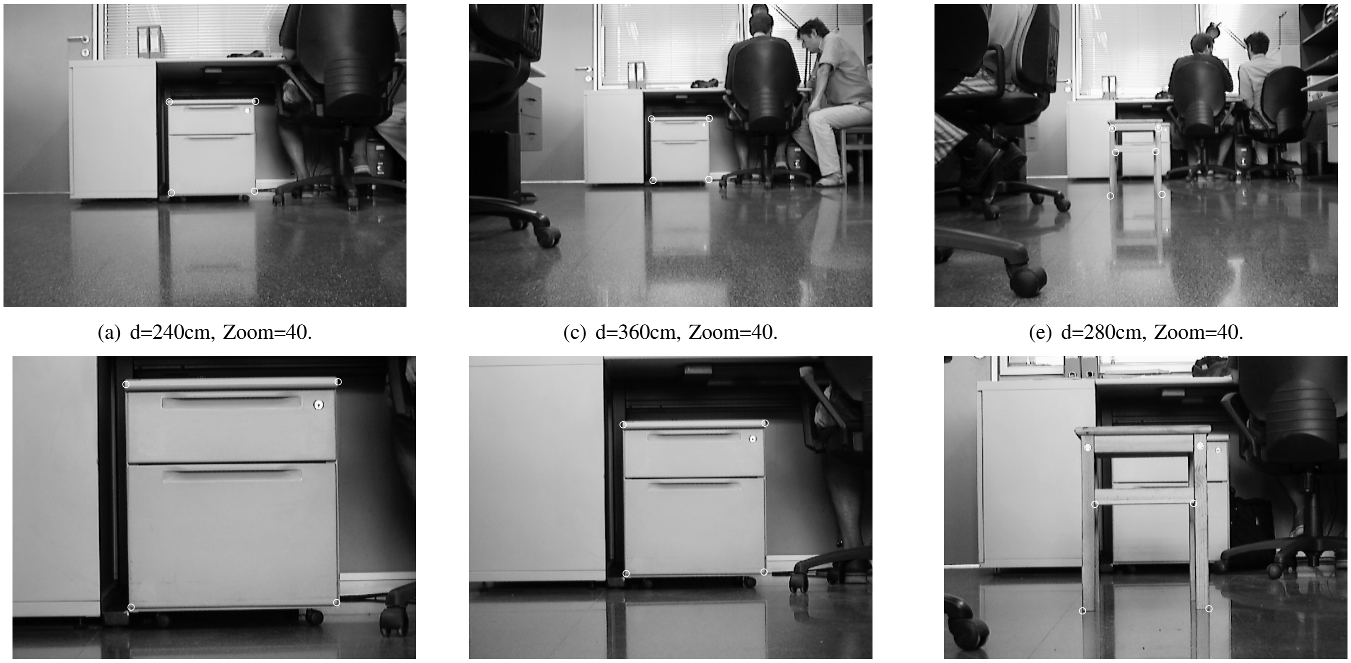
The robot takes an image pair with zoom in positions 40 and 708, at distances 240, 280, 320, 360 and 400 cm with respect of the table drawers. From Figure 4(a) to Figure 4(d) the image pairs corresponding to 240 and 360 cm are plotted. For the distance 280 we use also a wood stool (see Fig. 4(e) and 4(f)) to validate that the proposed method is only dependent on the zooming camera, and not on the calibration object. The idea is to perform the calibration off-line with a natural landmark, and use this calibration in real-time operations with any given new landmark, as usual with other calibration methods. The steps to compute the unknown depth are detailed in Alg 1.

```
for i=1 to 2 do
    Place camera at distance $d_i$ from the calibration object
    Compute the shape vector $S_i$ produced by the deformation between the image taken at $zoom_1$ and the one at $zoom_2$
end
Place the camera at unknown distance from the target object
Compute the shape vector $S$ produced by the deformation between the image taken at $zoom_1$ and the one at $zoom_2$
With $S_1, S_2$ and $S$ find the unknown distance by applying (7)
```

Algorithm 1: Steps of the depth estimation algorithm

Four points are manually extracted for each drawer image in order to construct the corresponding shape vector. For the stool images, six points are extracted instead, in order to assess the robustness of the shape vector obtained. As the method to obtain the shape vector through pseudoinverse multiplication can be seen as a minimization [19], the more point location measures are available, the more precision can be obtained.

Some results are summarized in Table I. The columns labelled Cal1 and Cal2 indicate the two distances used to perform the geometric calibration, and the other two columns show the estimated distance by the presented algorithm and



(a) d=240cm, Zoom=40.

(c) d=360cm, Zoom=40.

(e) d=280cm, Zoom=40.

(b) d=240cm, Zoom=708.

(d) d=360cm, Zoom=708.

(f) d=280cm, Zoom=708.

Fig. 4. For each camera position two images are needed to estimate the scale factor. The shown images correspond to the experiment labelled 4 in Table I. (a) (b) First calibration pair. (c)(d) Second calibration pair. (e)(f) Testing pair. Note that the calibration object and the one used for testing are not the same, and also different numbers of location measures are used to estimate the shape vector, 4 for the drawer images and 6 for the stool ones.

the measured one. For the experiments labelled 1 and 2, the camera is placed at 280 and 320 cm from the drawer, respectively. These depths are between the two calibration distances (240 and 360 cm), and the estimated depth is correctly computed by the algorithm in each experiment. In the experiment labelled 3 the camera is placed farther than the second calibration distance (out of the calibration range), and the depth is also recovered with small error, compared to the measured one. With these calibration parameters we perform a fourth experiment (numbered 4) using the 6 points extracted from the stool images. In this case depth is also reasonably recovered, although worse than in the previous cases.

In experiments 5, 6 and 7 the calibration range is shortened, using the calibration distances 240 and 320 cm. When the distance is between the calibration ones, as in experiment 5, the error is of the same order as in the previous experiment. When the camera is located farther than the second calibration distance, the depth is correctly recovered but with more error, compared to experiment 3. As typical in geometric calibration, the depth is correctly recovered within the range defined by the first and the second calibration distances as the algorithm is interpolating. Out of this zone the depth can be also inferred extrapolating, but the error grows as the distance increases. We find also that the larger the distance between calibration positions the more precision is obtained.

Finally, with experiment 8 we test the effect of moving both calibration camera positions farther away. Calibration was done with images taken at 320 and 400cm. A test is performed placing the camera in the middle obtaining a correct recovered depth.

V. CONCLUSIONS AND FUTURE WORKS

We have presented a simple method to determine the depth of a robot placement with respect to a landmark. The image deformation caused by zooming is modelled by a 3 degrees of freedom shape vector in a presented shape space, where the third element is the scale of the associated homotecy. This simple scale value is recorded at each calibration step. When a new scale is computed from the zooming of a new object, it can be compared to the calibration scales and, knowing the depth of the calibration objects, deduce the depth of the current target with a simple operation.

A minimum set of 3 point correspondences are needed to construct the affinity, but more correspondences will result in a better shape vector estimation, as a minimisation process is used. Here we have presented experiments using 4 and 6 correspondences between zooming images.

With the experiments we have demonstrated the validity of the method. The distance between calibration positions determines a calibrated zone where the algorithm is more precise. Out of this zone the algorithm also infers the depth but is less precise as the distance increases. We have demonstrated that the required shape vector can be calculated from different objects and using different numbers of point correspondences.

We have observed that the zooming sometimes drops the target out of the image. For practical purposes it is convenient to calibrate with some different zoom positions to be able to find one zoom range that contains the target in both images and for which we have calibration information.

Our objective has been mainly to remove from the ego-motion algorithm the scaling uncertainty, common in all monocular systems. But this method can be used also for other purposes, for instance, the initialisation of the pan and tilt controllers of our active vision system. Experiments with the PTZ control show that the obtained precision is enough to initialize the controllers in a good response zone.

In [2] we estimate the initial distance with a laser, and in [23] with a calibration pattern. Several other algorithms could benefit from the estimation of the initial distance of a given landmark. Let us just enumerate a few. Davison [24] estimate the depth of a landmark in monocular vision using a particle filter. In order to acquire the scale of the scene in the first frame a known object is used. Our method can be used thus changing the known object by any object in the scene. Sola [25] proposed to solve the depth initialisation problem with an approximation of the Gaussian Sum Filter, and Jensfelt et. al. [26] proposed to exclude from the SLAM process those features for which the depth had not been determined. When little disparity between matched features is present, for example in approaching robot motions and distant targets, all these methods could not extract significant information.

Recently Caballero et. al. [27] presented a monocular visual odometer for aerial vehicles. They proposed to measure the distance between the camera and the various targets used in the experiments with a sonar or a laser range sensor, but finally they did it manually.

Obviously, for traditional point-based maps it is not practical to perform the zoom positioning for each landmark initialisation. However, the presented algorithm is useful for those situations where an average depth is needed, as those mentioned before.

REFERENCES

[1] E. Martínez and C. Torras, “Qualitative vision for the guidance of legged robots in unstructured environments,” *Pattern Recognition*, vol. 34, pp. 1585–1599, 2001.

[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 *Proc. IEEE Int. Conf. Robot. Automat. (ICRA'05)*, Barcelona, Apr. 2005, pp. 3539–3544.

[3] E. Martínez and C. Torras, “Contour-based 3d motion recovery while zooming,” *Robotics and Autonomous Systems*, vol. 44, pp. 219–227, 2003.

[4] M. Li and J.-M. Lavest, “Some aspects of zoom-lens camera calibration,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 11, pp. 1105–1110, November 1996.

[5] J. A. Fayman, O. Sudarsky, E. Rivlin, and M. Rudzsky, “Zoom tracking and its applications,” *Machine Vision and Applications*, vol. 13, no. 1, pp. 25 – 37, 2001.

[6] B. Tordoff, “Active control of zoom for computer vision,” Ph.D. dissertation, University of Oxford, 2002.

[7] Y. Y. Schechner and N. Kiryati, “Depth from defocus vs. stereo: How different really are they?” *Int. J. Comput. Vision*, vol. 39, no. 2, pp. 141–162, 2000.

[8] J. Ma and S. I. Olsen, “Depth from zooming,” *J. Opt. Soc. Am. A*, vol. 7, no. 10, pp. 1883–1890, oct 1990.

[9] J.-M. Lavest, G. Rives, and M. Dhome, “Three-dimensional reconstruction by zooming,” *IEEE Trans. Robot. Automat.*, vol. 9, pp. 196–206, 1993.

[10] V. Rodin and A. Ayache, “Axial stereovision: Modelization and comparison between two calibration methods.” in *Proc. 1stIEEE Int. Conf. Image Process.*, Austin, Texas, Nov. 1994, pp. 725–729.

[11] J. Lavest, C. Delherm, B. Peuchot, and N. Daucher, “Implicit reconstruction by zooming,” *Comput. Vis. Image Und.*, vol. 66, no. 3, pp. 301–315, June 1997.

[12] H. A. Martins, J. R. Birk, and R. B. Kelley, “Camera models based on data from two calibration planes,” *Comp. Graph. Image Processing*, vol. 17, no. 2, pp. 173–180, 1981.

[13] K. Gremban, C. Thorpe, and T. Kanade, “Geometric camera calibration using systems of linear equations,” in *Proc. Image Understanding Workshop*, Cambridge, 1988, pp. 820–825.

[14] C. Delherm, J.-M. Lavest, M. Dhome, and J.-T. Laprest, “Dense reconstruction by zooming,” in *Proc. 4th European Conf. Comput. Vision*, ser. Lect. Notes Comput. Sci., B. Buxton and R. Cipolla, Eds., vol. 1065. London, UK: Springer-Verlag, Apr. 1996, pp. 427–438.

[15] M. Baba, N. Asada, A. Oda, and T. Migita, “A thin lens based camera model for depth estimation from blur and translation by zooming,” in *Proc. 15th Int. Conf. Vision Interface*, Calgary, May 2002, pp. 274–281.

[16] Z. Myles and N. da Vitoria Lobo, “Recovering affine motion and defocus blur simultaneously,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 6, pp. 652–658, June 1998.

[17] J. Koenderink and A. J. van Doorn, “Affine structure from motion,” *J. Opt. Soc. Am. A*, vol. 8, no. 2, pp. 377–385, 1991.

[18] L. S. Shapiro, A. Zisserman, and M. Brady, “3D motion recovery via affine epipolar geometry,” *Int. J. Comput. Vision*, vol. 16, no. 2, pp. 147–182, 1995.

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

[20] J. Foley, A. van Dam, S. Feiner, and F. Hughes, *Computer Graphics. Principles and Practice*. Addison-Wesley Publishing Company, 1996.

[21] E. Martínez, “Recovery of 3d structure and motion from the deformation of an active contour in a sequence of monocular images,” Ph.D. dissertation, Universitat Politï¿½nica de Catalunya, 2000.

[22] B. Tordoff and D. Murray, “Reactive control of zoom while fixating using perspective and affine cameras,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, no. 1, pp. 98–112, January 2004.

[23] M. Alberich-Carramiñana, G. Alenyà, J. Andrade-Cetto, E. Martínez, and C. Torras, “Affine epipolar direction from two views of a planar contour,” in *Advanced Concepts for Intelligent Vision Systems*, ser. Lect. Notes Comput. Sci., vol. 4179, Antwerp, Sep. 2006, pp. 944–955.

[24] A. Davison, “Real-time simultaneous localisation and mapping with a single camera,” in *Proc. IEEE Int. Conf. Comput. Vision*, Nice, Oct. 2003, pp. 1403–1410.

[25] J. Sola, A. Monin, M. Devy, and T. Lemaire, “Undelayed initialization in bearing only SLAM,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Edmonton, Aug. 2005.

[26] P. Jensfelt, D. Kragic, J. Folkesson, and M. Bjorkman, “A framework for vision based bearing only 3d slam,” in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA'06)*, Orlando, May 2006, pp. 1944–1950.

[27] F. Caballero, L. Merino, J. Ferruz, and A. Ollero, “A visual odometer without 3d reconstruction for aerial vehicles. applications to building inspection,” in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA'05)*, Barcelona, Apr. 2005, pp. 4673–4678.

Computation of Rotation Local Invariant Features using the Integral Image for Real Time Object Detection

Michael Villamizar[1], Alberto Sanfeliu[1] and Juan Andrade-Cetto[2]
[1] *Institut de Robòtica i Informàtica Industrial, CSIC-UPC*
[2] *Computer Vision Center, Universitat Autònoma de Barcelona*

Abstract

We present a framework for object detection that is invariant to object translation, scale, rotation, and to some degree, occlusion, achieving high detection rates, at 14 fps in color images and at 30 fps in gray scale images. Our approach is based on boosting over a set of simple local features. In contrast to previous approaches, and to efficiently cope with orientation changes, we propose the use of non-Gaussian steerable filters, together with a new orientation integral image for a speedy computation of local orientation.

1. Introduction

Object detection is a fundamental issue in most computer vision tasks; particularly, in applications that require object recognition. Early approaches to object recognition are based on the search for matches between geometrical object models and image features. Appearance-based object recognition gained popularity in the past two decades using dimensionality reduction techniques such as PCAs for whole-image matching. Lately, a new paradigm for object recognition has appeared based on the matching of geometrical as well as appearance local features. Moreover, the use of boosting techniques for feature selection has proven beneficial in choosing the most discriminant geometric and appearance features from training sets.

In this paper we focus on the selection of local features invariant to translation, scaling, orientation, and to some degree, occlusion. Our approach differentiates from others in that while based on boosting over a set of training samples, it can achieve object detection in real time. This is thanks to our extension of the use of steerable filters to non-Gaussian kernels, together with our proposal of a new integral image for the computation of local image orientation.

Viola and Jones [10] introduced the integral image for very fast feature evaluation. Once computed, the integral image allows the computation of Haar-like features [5] at any location or scale in real time. Unfortunately, such system is not invariant to object rotation or occlusions.

Other recognition systems that might work well in cluttered scenes are based on the computation of multi-scale local features such as the SIFT descriptor [3]. One key idea behind the SIFT descriptor is that it incorporates canonical orientation values for each keypoint. Thus, allowing scale and rotation invariance during recognition. Even when a large number of SIFT features can be computed in real time for one single image, their correct pairing between sample and test images is performed via nearest neighbor search and generalized Hough transform voting, followed by the solution of the affine relation between views; which might end up to be a time consuming process.

Yokono and Poggio [11, 12] settle for Harris corners at various levels of resolution as interest points, and from these, they select as object features those that are most robust to Gaussian derivative filters under rotation and scaling. As Gaussian derivatives are not rotation invariant, they use steerable filters [1] to steer all the features responses according to the local gradient orientation around the interest point. In the recognition phase, the system still requires local feature matching, and iterates over all matching pairs, in groups of 6, searching for the best matching homography, using RANSAC for outlier removal. Unfortunately, the time complexity or performance of their approach was not reported.

Work by many others is also related to the issue of rotation invariant feature matching [4]. We feel however, the success of our approach to be founded on the ideas presented in the former three contributions: boosting, canonical orientation, and steerable filters, along with the intro-

Financial support to M. Villamizar and A. Sanfeliu comes from the EURON Network Robot Systems Research Atelier NoE-507728, and the Spanish Ministry of Education and Science project NAVROB DPI 2004-05414. J. Andrade-Cetto is a Juan de la Cierva Postdoctoral Fellow of the Spanish Ministry of Education and Science under project TIC2003-09291, and is also funded in part by the EU PACO-PLUS project FP6-2004-IST-4-27657. The authors belong to the Artificial Vision and Intelligent Systems Group funded in part by the Catalan Research Commission. We thank Joan Purcalla for initial AdaBoost code.

duction in this paper of the integral image for orientations, and its extension to non-Gaussian steerable filters.

In our system, keypoints are chosen as those regions in the image that have the most discriminant response under convolution with a set of wavelet basis functions at several scales and orientations. Section 2 explains how the most relevant features are selected. The selection is made with a boosting mechanism, producing a set of weak classifiers and their corresponding weights. A linear combination of these week classifiers produces a strong classifier, which is used for object detection. Rotation invariance is achieved by filtering with oriented basis functions. Filter rotation is efficiently computed with the aid of a steerable filter [1], that is, as the linear combination of basis filters, as indicated in Section 3.

During the recognition phase, sample image regions must be rotated to a trained canonical orientation, as explained in Section 4, prior to feature matching. Such orientation is dictated by the peak on a histogram of gradient orientations, depicted in Section 5. One of the major contributions of this paper is the efficient computation of image region orientation by means of an integral image of gradient orientation histograms; enabling our system to perform object detection invariant to translation, scaling, orientation, and some degree of occlusion, in real time. Section 6 is devoted to some experimental results of the overall approach, and Section 7 has some concluding remarks.

2. Feature Selection

The set of local features that best discriminates an object is obtained by convolving positive sample images with a simplified set of wavelet basis function operators [5] at different scales and orientations. These filters have spatial orientation selectivity as well as frequency selectivity, and produce features that capture the contrast between regions representing points, edges, and strips, and have high response along for example, contours. The set of operators used is shown in Figure 1. Filter response is equivalent to the difference in intensity in the original image (or color channel magnitude) between the dark and light regions dictated by the operator.

Convolving these operators at any desired orientation is performed by steering the filter (Section 3). Furthermore, fast convolution over any region of the entire image is efficiently obtained using an integral image (Section 5).

Feature selection is performed via a boosting mechanism, namely, AdaBoost [2]. AdaBoost extracts in each iteration the weak classifier (filter width, location, type, orientation, and threshold) that best discriminates positive from negative training images. A weak classifier can be expressed as

$$h(I) = \begin{cases} 1 & : \quad I * f > t \\ 0 & : \quad \text{otherwise} \end{cases} ,$$

where I is a training sample image, f is the filter being tested, with all its parameters (width, location, type, and orientation), $*$ indicates the convolution operation, and t is the filter response threshold. The algorithm selects the most discriminant weak classifier, as well as its contribution α in classifying the entire training set, as a function of the classification error ϵ; $\alpha = \frac{1}{2} \ln \frac{1-\epsilon}{\epsilon}$.

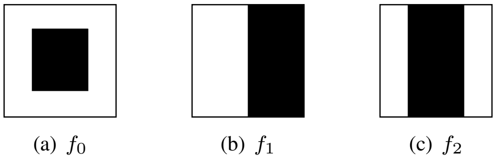

(a) f_0 (b) f_1 (c) f_2

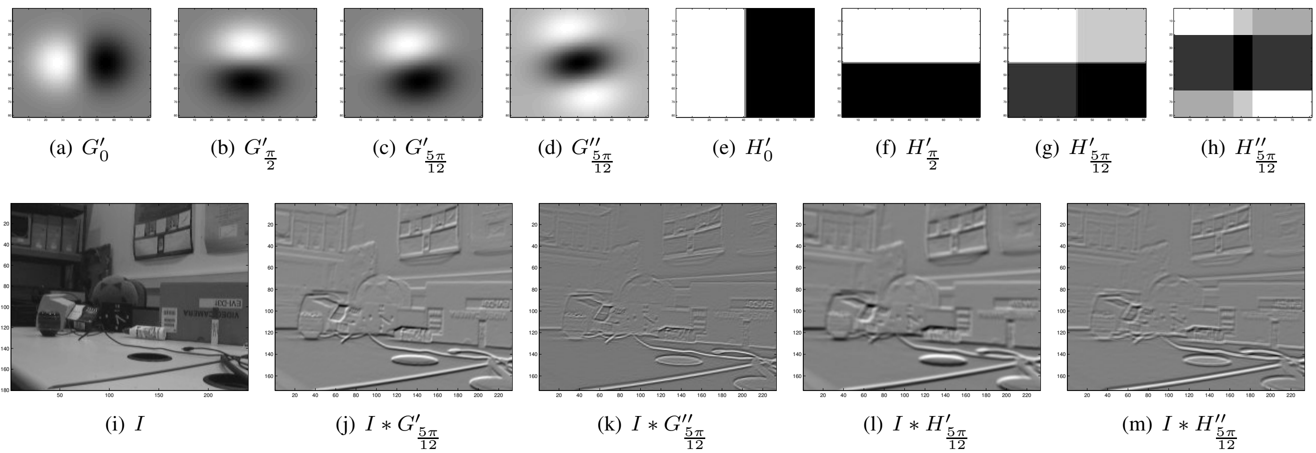
Figure 1. Simplified wavelet basis function set. a) center-surround b) edge, and c) line.

At each iteration, the algorithm also updates a set of weights over the training set. Initially, all weights are set equally, but on each round, the weights of missclassified samples are increased so that the algorithm is forced to focus on such hard samples in the training set the previously chosen classifiers missed. In a certain way, the technique is similar to a Support Vector Machine, in that both search for a class separability hyperplane, although using different distance norms, l_2 for SVMs, and l_1 for boosting [7]. The dimensionality of the separating hyperplane in AdaBoost is given by the number N of weak classifiers that form the strong classifier

$$H(I) = \begin{cases} 1 & : \quad \sum^N \alpha_i h_i(I) \geq \frac{1}{2} \sum^N \alpha_i & \text{object} \\ 0 & : \quad \text{otherwise} & \text{no-object} \end{cases} .$$

To achieve invariance to translation during the detection phase, the strong classifier H is tested for a small window the size of the training samples (30×30 pixels), and at every pixel for the entire test image. To speed up the process, the test can be performed every two or three pixels (or rows), with the compromise of possibly missing the object, i.e., having a false negative. In practice, this increment can be made up to 10% the size of the training sample, without incurring in false negatives.

Similarly, scale invariance is obtained by scaling each filter within the classifier H. Scaling of the filters can be performed in constant time for a previously computed integral image. Our tests show that we can scale up to 20% the size of the training sample, with still good detection rates.



(a) G'_0 (b) $G'_{\frac{\pi}{2}}$ (c) $G'_{\frac{5\pi}{12}}$ (d) $G''_{\frac{5\pi}{12}}$ (e) H'_0 (f) $H'_{\frac{\pi}{2}}$ (g) $H'_{\frac{5\pi}{12}}$ (h) $H''_{\frac{5\pi}{12}}$

(i) I (j) $I * G'_{\frac{5\pi}{12}}$ (k) $I * G''_{\frac{5\pi}{12}}$ (l) $I * H'_{\frac{5\pi}{12}}$ (m) $I * H''_{\frac{5\pi}{12}}$

Figure 2. First and second order Gaussian and wavelet-based steerable filterss. (a-b) and (e-f) basis, (c-d) and (g-h) oriented filters, (i) original image, (j-m) filter responses.

3. Steerable Filters

In order to achieve orientation invariance, the local filters must be rotated previous to convolution. A good alternative is to compute these rotations with steerable filters [1], or with its complex version [8]. A steerable filter is a rotated filter comprised of a linear combination of a set of oriented basis filters, $I * f(\theta) = \sum^n k_i(\theta) I * f(\theta_i)$, where $f(\theta_i)$ are the oriented basis filters, and k_i are the coefficients of the bases.

Consider for example, the Gaussian function $G(u, v) = e^{-(u^2+v^2)}$, and its first and second order derivative filters $G'_u = -2ue^{-(u^2+v^2)}$ and $G''_u = (4u^2-2)e^{-(u^2+v^2)}$. These filters can be re-oriented as a linear combination of filter bases. The size of the basis is one more than the derivative order.

Consequently. the first order derivative of our Gaussian function at any direction θ, is $G'_\theta = \cos\theta G'_u + \sin\theta G'_v$, and a steered 2nd order Gaussian filter is obtained with $G''_\theta = \sum_{i=1}^3 k_i(\theta) G''_{\theta_i}$, with $k_i(\theta) = \frac{1}{3}(1+2\cos(\theta-\theta_i))$; and G''_{θ_i} the precomputed second order derivative kernels at $\theta_1 = 0$, $\theta_2 = \frac{\pi}{3}$, and $\theta_3 = \frac{2\pi}{3}$.

Convolving with Gaussian kernels is a time consuming process. Instead, we propose to approximate such filter response by convolving with the Haar basis from Figure 1. This, with the aid of an integral image. $I * f_1(\theta) = \cos\theta I * f_1(0) + \sin\theta I * f_1(\frac{\pi}{2})$. Similarly, filtering with our line detector at any orientation θ is obtained with $I * f_2(\theta) = \sum_{i=1}^3 k_i(\theta) I * f_2(\theta_i)$.

The similarity of the response to Haar filters allows us to use this basis instead as weak classifiers for the detection of points, edges, and lines; just as the Gaussian filters do. The main benefit of the approach is in speed of computation. While convolution with a Gaussian kernel takes time O(n) the size of the kernel, convolution with the oriented Haar basis can be computed in constant time using an integral image representation. Figure 2 shows the results of the proposed feature selection process.

4. Local Orientation

Say, a training session has produced a constellation H of local features h as the one shown in Figure 4. Now, the objective is to test for multiple positions and scales in each new image, whether such constellation passes the test H or not. Instead of trying every possible orientation of our constellation, we chose to store the canonical orientation θ_0 of H from a reference training image block, and to compare it with the orientation θ of each image block being tested. The difference between the two indicates the amount we must re-orient the entire feature set before the test H is performed.

$$\psi = \begin{cases} \theta - \theta_0 & : \quad \theta \geq \theta_0 \\ \theta - \theta_0 + 2\pi & : \quad \text{otherwise} \end{cases}$$

On way to compute block image orientation is with ratio of first derivative Gaussians G'_u and G'_v [12], $\tan\theta = \frac{I * G'_v}{I * G'_u}$.

Another technique, more robust to partial occlusions, is to use the mode of the local gradient orientation histogram (see Figure 4), for which it is necessary to compute gradient orientations pixel by pixel, instead of a region convolution as in the previous case. When the scene is highly structured, such histogram can easily be multimodal. We follow for such cases the same convention as with SIFT features: for any peak in the histogram greater than 80% the size of the

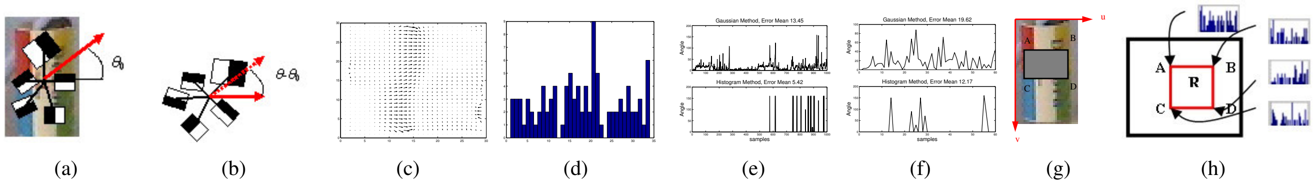


Figure 3. Image orientation computed as the mode of the gradient orientation image. a) canonical orientation, b) rotated constellation, c) image gradients, d) gradient orientation histogram, local orientation error subject to e) scale change, and f) small occlusions, g) integral image, and h) local histogram integral image.

mode, a new weak classifier, oriented at that value is added to the classifier set.

We have done several tests to estimate which of these two techniques for computing local image orientation is most suitable to our needs. As shown in Figure 3(e), computing local orientation using the histogram deteriorates more with scale changes than computing the gradient over the entire image block. However, as seen in Figure 3(f), given that the mode is a nonlinear filter, the technique is much more reliable in the presence of small occlusions. We settle for the histogram mode to handle occlusions, and let the boosting mechanism deal with translation and scale affinities.

5. The Local Orientation Integral Image

An integral image is a representation of the image that allows a fast computation of features because it does not work directly with the original image intensities (color values). Instead, it works over an incrementally built image that adds feature values along rows and columns. Once computed this image representation, any one of the local features (weak classifiers) can be computed at any location and scale in constant time.

In its most simple form, the value of the integral image M at coordinates u, v contains the sum of pixels values above and to the left of u, v, inclusive, $M(u, v) = \sum_{i \leq u, j \leq v} I(i, j)$,

Then, it is possible to compute for example, the sum of intensity values in a rectangular region simply by adding and subtracting the cumulative intensities at its four corners in the integral image, Area $= A + D - B - C$.

Furthermore, the construction of the integral image is $O(n)$ in the size of the image, and is computed iteratively with $M(u, v) = I(u, v) + M(u-1, v) + M(u, v-1) - M(u-1, v-1)$.

In this form, the response from the two orthogonal Haar-filter basis from Figure 2, at any size or location, can be computed by simple adding and subtracting four values from the integral image. This, in constant time.

Extending the idea of having cumulative data at each pixel in the Integral Image, we decide to store in it orientation histogram data instead of intensity sums. Once constructed this orientation integral image, it is possible to compute a local orientation histogram for any given rectangular area within an image in constant time. Histogram(Area) $=$ Histogram(A) $+$ Histogram(D) $-$ Histogram(B) $-$ Histogram(C).

6. Experiments

For the experiments reported here, our training set had 5250 negative images and 1100 positive images. Negative images were obtained under varying illumination conditions, both from exterior and interior scenes. In order to have the boosting mechanism choose the most invariant classifiers, we have added as positive samples, synthetic images where the object to be learned appears translated, rotated, and scaled. Object translations reach 5 pixels in all directions. Scaling of the object images goes up to 20% of the original image size, and rotated images reach 10 degrees in order to aid the histogram method which was chosen to have a precision of 10 degrees, given that has 36 bins. Some positive and negative samples are shown in Figure 4.

Figure 5 shows some frames of a sequence in which the trained object is being recognized. At some point, the object is being detected at multiple neighboring locations, fact indicated by the repetitive superimposed squares. Frame (a) shows the object being detected as trained; frames (b-d) show robustness to orientation changes; frame (e) shows detection at a different scale; frames (f) and (g) show detection at both different scale and orientation; and frame (h) shows positive detection under scale, orientation, and mild occlusion.

Note however, that while convolution with the two orthogonal basis required for the first order Haar filter can be computed using an integral image; the same is not true for the second order filter since it requires basis kernels oriented at $\frac{\pi}{3}$ rad. and $\frac{2\pi}{3}$ rad., besides the already orthogonal basis

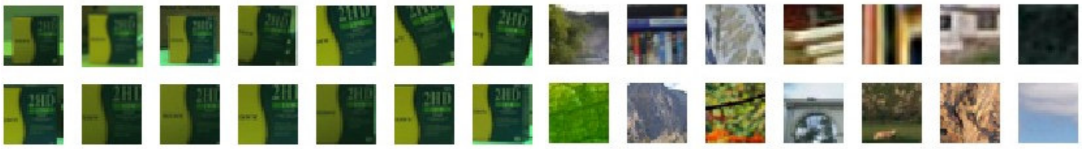
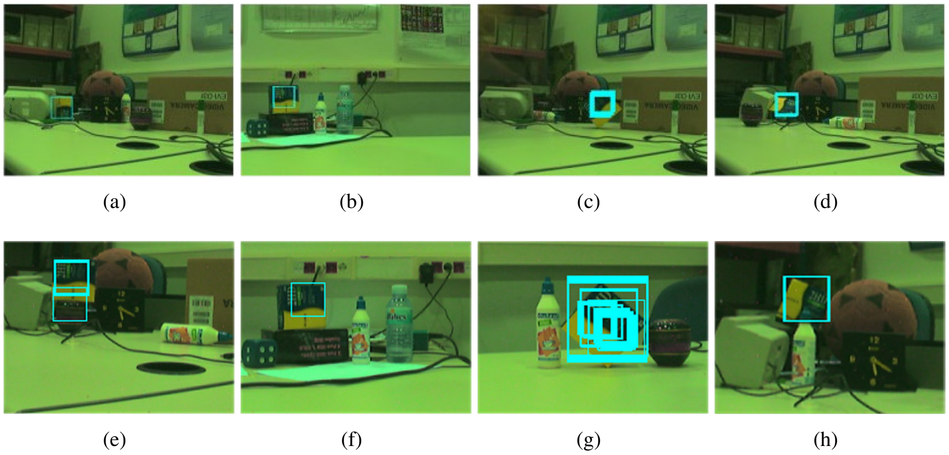


Figure 4. Positive and negative samples.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 5. Some frames that show the object being detected under varying scales, orientation, and mild occlusion.

at 0 rad. Fortunately, our experiments indicate that line features are seldom chosen by the boosting algorithm as weak classifiers, accounting in the worst cases for at most 20% the total number of weak classifiers, and in little detriment of speed of computation. Nevertheless, the computation of these basis kernels in a fast integral-image-like manner is a subject of further study.

7. Conclusions

In this paper we have presented a system for object detection that is invariant to object translation, scale, rotation, and to some degree, occlusion, achieving high detection rates, at 14 fps in color images and at 30 fps in gray scale images. Our approach is based on boosting over a set of simple local features. In contrast to previous approaches, and to efficiently cope with orientation changes, we propose the use of Haar basis functions and a new orientation integral image for a speedy computation of local orientation.

References

[1] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE T. PAMI*, 13(9):891–906, 1991.

[2] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, Aug. 1997.

[3] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[4] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE T. PAMI*, 27(10):1615–1630, Oct. 2005.

[5] C. P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Proc. IEEE ICCV*, page 555, Bombay, Jan. 1998.

[6] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE T. PAMI*, 20(1):23–38, 1998.

[7] G. Rätsch, B. Schölkopf, S. Mika, and K.-R. Müller. SVM and Boosting: One class. Tech. Rep., GMD First, Nov. 2000.

[8] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?". In *Proc. ECCV*, pages 414–431, Copenhagen, 2002.

[9] B. Schiele and J. L. Crowley. Object recognition using multidimensional receptive field histograms. In *Proc. ECCV*, pages 610–619, Cambridge, Apr. 1996.

[10] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. IEEE CVPR*, pages 511–518, Kauai, Dec. 2001.

[11] J. Yokono and T. Poggio. Oriented filters for object recognition: An empirical study. In *Proc. 6th IEEE Int. Conf. Automatic Face Gesture Recog.*, pages 755–760, Seoul, 2004.

[12] J. Yokono and T. Poggio. Rotation invariant object recognition from one training example. Tech. Rep. 2004-010, MIT AI Lab., Apr. 2004.