BUILDING A CONSISTENT 3D REPRESENTATION OF A MOBILE ROBOT ENVIRONMENT BY COMBINING MULTIPLE STEREO VIEWS 1

Nicholas AY ACHE and Olivier D. FAUGER AS

INRIA - Domaine de Voluceau - Rocquencourt - BP. 105 - 78153 LE CHESNAY Cedex - FRANCE

ABSTRACT

In this short article, we report new results on our work on the problem of using passive Vision and more precisely Stereo Vision to build up consistent 3D geometric descriptions of the environment of a mobile robot

I-INTRODUCTION

The robot that we have built consists of a four-wheeled platform with two driving wheels operated by electrical motors. A set of three CCD cameras provides black and white images of the environment The cameras are located at the vertexes of a vertical roughly equilateral triangle. Images are transmitted via a VHF link to a workstation where they are stored and made available through Ethernet to a number of processors. To the user, the vehicle appears as a standard peripheral and can be accessed as such from any terminal on the net It is therefore a very convenient testbed for studying a number of problems in Vision.

One such problem is the following. Suppose we let our vehicle wander around in a building using its ultrasound sensors to avoid obstacles, odometry to roughly estimate its motion and its three cameras to compute 3D descriptions of its environment One question then is, can we hope to combine coherently the various sources of information, and especially the visual information obtained at different times and from different places, and build up an accurate geometric 3D representation of the building even if each individual measurement is itself fairly inaccurate 7 We call this problem the *Visual Fusion problem.*

There are two deep issues which are associated with this question. First is the issue of the type of geometric representation that is used by the system. Representations which are mathematically equivalent may behave quite differently on a real problem due to the unavoidable presence of noise and errors. This brings up the second issue which is the question of how do we represent and manipulate uncertainty.

In the next Sections we propose a solution to these issues and present some results.

II - WHAT IS THE PROBLEM THAT WE ARE TRYING TO SOL VE ?

Each triplet of images provided by the three cameras is analysed by a Stereo program described in [3,4]. This program outputs 3D line segments described in a coordinate system attached to the three cameras. Each line segment has a geometric description which we elaborate on in the next Section and an uncertainty which we explain in Section IV. This uncertainty is directly related to the limited resolution and the geometry of the three cameras.

To relate the various coordinate systems corresponding to the different viewpoints we estimate the rigid motions between them. This is done in two steps. First a rough estimate is obtained by combining the odometry with the rotation of the cameras. Second, a better estimate is obtained by combining the two 3D representations provided by the Stereo program in the two positions of the vehicle. This is done by matching 3D segments which are present in the two views and is described in details in [5,1]. The result is an estimate of the rotation matrix and translation vector between the coordinate systems attached to the cameras in their respective positions, together with some measure of their uncertainty (to be explained in Section IV).

This having been completed, the current representation of the environment is a number of uncertain geometric primitives (here 3D line segments) attached to coordinate frames related by uncertain rigid motions. The more we move the robot and measure, the move we increase the number of line segments until we run out of memory. This is clearly unsatisfactory and we must provide the system means of "forgetting intelligently". By this we mean the following. Let us consider a physical line segment S like a part of the frame of a window, or the edge of a desk. This line segment is very likely to have been detected in different positions 1, 2, ... , n of the mobile robot and is therefore present as segment Sj in position 1, segment S_2 in position 2, ... , segment S_n in position n. Since we can relate by rigid motions position 1, to positions 2, 3. . . . .n, by applying the right transformation, the physical segment S is represented by n segments Sj, S'_2,... , S'_n in the coordinate system attached to position 1.

We would like our system to have the capability of automatically deciding that S_1, S_2 ... ♦ and S'_n are the same segment S and fusing them into one segment *S* , a combination of Sj, S_2, ... , S'_n. This has two advantages. First, *S* being a combination of n sources of information should be at least as accurate as each of its instanciations, therefore accuracy in the description is now able to increase, and second since the system has recognized that Sj, S_2, . . . , and S_n are the same segment *S* , it can forget diem and remember only *S*, it is now able to "forget intelligently".

In order to achieve this goal, two questions must be answered. How do we represent and manipulate geometry and uncertainty ?

HI - REPRESENTING LINES AND LINE SEGMENTS

The obvious way to represent a line is by choosing two points on it, or one point and a direction. The first representation has dimension 6 (the six coordinates of the two points), the second representation has dimension 5 (the three coordinates of the point and the two coordinates defining die direction as, for example a unit vector on the gaussian sphere). In fact, the minimal dimension of the representation of a line is four. This can be seen by choosing, in the second representation, the point such mat the segment from the origin to the point is perpendicular to the line. The line is then located in the plane normal to that segment and can be determined by its orientation with respect to a known direction in that plane, i.e. by one parameter.

A line segment is six-dimensional, being represented either by its two endpoints or by one endpoint (3 parameters), the line direction (2 parameters), and its lengm (1 parameter).

For our problem, even though we actually manipulate line segments because this is what is provided by our stereo algorithms, what we in fact would like to fuse are lines. The reason for this is the fact mat segmentation errors, variations of illumination, inadequate edge detectors, and variations of viewpoints result in the same physical segments being instantiated as a variety of subsegments. Since we do not know the real segment we must deal with its supporting line.

A convenient minimal (i.e. four-dimensional) representation of 3D lines is the (a, b, p, q) representation where the line is defined by the two planes:

$$x \bullet az + p \qquad y - bz + q \qquad (1)$$

This representation is easily computed using the coordinates of two points on the line.

The effect of a rigid motion defined by a rotation matrix R and a translation vector t can be readily assessed, i.e. if we route and translate a

1: This work is supported by the ESPRIT project P940.

line L into a line L, a', b', p: and q' can be easily computed as functions of a, b, p, q, R, and t. The details of the computation can be found in [2].

IV - REPRESENTING GEOMETRIC UNCERTAINTY

Uncertainty cannot be engineered away, therefore it has to be present explicitly in the representation that is manipulated by the Vision system. In our example, we have two types of uncertainty. First, the uncertainty on the localization of the 3D segments in each coordinate frame and second, the uncertainty on the rigid motions relating the various coordinate frames.

The stereo reconstruction program relates the pixel uncertainty in the three images to the uncertainty of the endpoints of the 3D segments as covariance matrixes. From them, we derive the covariance matrix of the (a, b, p, q) representation of the supporting 3D line (see [2]).

The rigid motions between coordinate frames are represented by two three-dimensional vectors, r representing the rotation and t the translation. The uncertainty on the rigid motions is then represented as covariance matrixes on those vectors. For details on the representation of rigid motions and the computation of their uncertainty, the interested reader is referred to [5,1,2].

The key assumption underlying these computations is that the various geometric representations that we manipulate are well modelled by gaussian processes (thus the use of covariance matrixes to represent their uncertainty). It is impossible to demonstrate theoretically the validity of this assumption, the quality of the results that we show in Section V is a practical confirmation of it. This in turn allows us to use the extremely powerful tool of the Extended Kalman Filtering (EKF) to manipulate this uncertainty. Due to lack of space, we cannot develop here the corresponding formalism which the interested reader can find in some of the previous references.

Let us now see how we can use these tools to solve our problem. We treat the case of a segment S1 in coordinate frame 1 and a segment S_2 in coordinate frame 2 related by a rigid transformation T (if we apply T to segments in frame 2 they are expressed in frame 1). We first discuss the case where the segments are represented by their endpoints and then the case of the minimal representation defined by equations (1). In both cases, T has uncertainty defined by a covariance matrix A.

IV.1 - Fusing segments endpoints

Segment S_1 is represent in frame 1 by its endpoints M_1 and P_1 and segment S_2 in frame 2 by its endpoints M_2 and P_2. Each endpoint has also a covariance matrix, i.e. Λ_{M_1} for M_1, etc ... .

Applying the rigid motion T to S_2 yields a segment S'_2 in frame 1 represented by its endpoints $M'_2 = T(M_2)$ and $P'_2 = T(P_2)$ and their covariance matrices $\Lambda_{M'_2}$, and $\Lambda_{P'_2}$, which can be computed by the EKF formalism from Λ_{M_2}, Λ_{P_2} and A.

Two questions arise at this point : 1 - are S_1 and S'_2 two instances of the same physical segment ; 2 - if yes, what is the representation of the fused segment S ?

The answer to the first question is provided by noticing that if M_1 and M'_2 are independent gaussian points, then the covariance matrix of the gaussian vector $M_1M'_2$ is the sum $\Lambda_{M_1} + \Lambda_{M'_2}$, of the covariance matrixes of its endpoints. If M_1 and M'_2 are also two instances of the same gaussian point M, then the exptected value of $M_1M'_2$ is 0 and the quantity:

$$d^2(M_1, M'_2) = M_1M'^T_2(\Lambda_{M_1} + \Lambda_{M'_2})^{-1} M_1M'_2$$

has a χ^2 distribution. Therefore we can fix a threshold s such that $d^2(M_1,M'_2)$ has a probability of, let us say 95 %, of being less than this threshold.

This is the test we use to decide whether M_1 and M'_2 are two instances of the same gaussian distributed point M : given M_1 and M'_2, and Λ_{M_1} and $\Lambda_{M'_2}$, we compute $d^2(M_1, M'_2)$; if it is less than s, then we decide that M_1 and M'_2 can be fused.

The answer to the second question is provided by the EKF formalism that yields the endpoints $M=Fusion(M_1,M'_2)$ and $P=Fusion(P_1,P'_2)$ of the segment S as well as their covariance matrices $\Lambda_M=Fusion(\Lambda_{M_1},\Lambda_{M'_2})$ and $\Lambda_P = Fusion(\Lambda_{P_1},\Lambda_{P'_2})$. Formulas can be found in the previously cited references.

This scheme will work well if segments are not broken too differently in the two views. In practice they are and we have developed a second scheme based on the representation described in Section III.

IV.2 - Fusing lines

An approach similar to the previous one can be developed for lines. Segment S_1 is represented in frame 1 by its supporting line L_1 itself represented by (a_1, b_1, p_1, q_1). Segment S_2 is represented in frame 2 by its supporting line L_2 itself represented by (a_2, b_2, p_2, q_2). Each line has also a covariance matrix Λ_{L_1} and Λ_{L_2}.

Applying the rigid motion T to L_2 yields a line L'_2 in frame 1 represented by (a'_2, b'_2, p'_2, q'_2) and a covariance matrix $\Lambda_{L'_2}$, which can be computed from Λ_{L_2} and A (the rigid motion covariance) by the EKF formalism.

The same questions asked in Section IV.1 can be answered similarly. L_1 and L'_2 are fused if $d^2(L_1,L'_2)$ is less than or equal to a threshold s corresponding to a 95 % probability of their being two instances of the same gaussian line. $d^2(L_1, L'_2)$ is computed as the Mahalanobis distance between their representations.

$$d(L_1, L'_2) = (L_1 L'_2)^T (\Lambda_{L_1} + \Lambda_{L'_2})^{-1} (L_1 L'_2)$$

where the four-dimensional vector $L_1 L'_2$ is equal to $(a_1-a'_2, b_1-b'_2, p_1-p'_2, q_1-q'_2)^T$.

If $d^2(L_1, L'_2)$ is less than the threshold then L_1 and L'_2 are fused into L defined by its representation (a,b,p,q) and its covariance matrix A, both being provided by the EKF formalism. The corresponding segment is obtained by projecting the endpoints M_1 and P_1 of S_1 and M'_2 and P'_2 of S'_2 on L and looking for maximal length connected components.

V - RESULTS

We have experimented the previous formalism both on synthetic and real data. Here are two typical examples.

First example is synthetic. We build a synthetic rigid object made of 8 vertical segments forming a square with a diamond in it. We form seven noisy instances of this object by the following technique : first, we generate a noisy displacement T on the object. This is done by generating a gaussian 6-vector of zero mean and small given covariance representing the parameters (r,t) of the rotation and translation. Second we modify the length of each segment by multiplying it by a random factor X uniformly distributed between 0.9 and 1.1. (This is to simulate the polygonal approximation instabilities.) Finally, we select a different covariance matrix for each endpoint, and apply a corresponding zero-mean gaussian noise. This is repeated seven times and the result is shown in figure 1, where the front view is above the vertical view. The ellipses are a representation of the covariance matrices attached to the endpoints, derived by the Kalman filter formalism to take into account the uncertainty on the motion and on the endpoints (but not the uncertainty on the segments lengths). To fuse these segments, we first applied a simple least squares algorithm between homologous endpoints. The result is shown in figure 2. We also applied the formalism developped in IV.1, i.e. fused two points when the Mahalanobis distance between them was lower than 7.8 (corresponding to a 95 % confidence rate) and applied a Kalman Filter to compute a better estimation. The result is shown in figure 3 with the a posteriori covariance matrices. Finally we did the same with the formalism developed in IV.2, fusing lines instead of points. The projection of the initial points on the resulting lines and the a posteriori covariance on them are shown in figure 4. The computed mean square error between the actual endpoints and the fused segments are respectively 4.3 %, 3.2 %, 1.1 % with respect to the side of the square, which shows the superiority of the Extended Kalman Filter over simple least-squares, and also the superiority of the line fusion

over the endpoint fusion. This last point is due the arbitrary segment breaking, which is not taken into account in the endpoint covariances, whereas it does not affect the line representations.

Actually the same experiment conducted without line breaking yielded a similar error rate of about 0.5 % for the two Kalman approaches, while the least-squares technique was still yielding a 4.S % mxerror.

The second experiment is conducted with real data. Figures 5 to 8 shows a calibration grid observed by our mobile robot from 4 different positions. Actually the robot has 3 eyes (cameras) and builds in each position a set of 3D segments. Covariance matrices are computed on the segments endpoints assuming gaussian noise in the images (see [1]). Using those segments which are common to a pair of successive views, the system computes the 3D motion between each such pair, and then places all the segments into a single reference frame. Figure 9 shows a vertical view of the grid segments in such a frame.

Fusing is then achieved between endpoints using the formalism described in IV.I. allowing a reduction of the spread of the vertical projection of the segments akmg 2 ideal lines (vertical segments lie slightly in front of horizontal ones). This spread agrees very well with the computed a priori and a posteriori computed covariance matrices (not shown in these figures).

VI - DISCUSSION AMD CONCLUSIONS

We have expanded in this paper on some key and simple motions which we have presented elsewhere [1,2).

First we have obtained more evidence of the necessity to combine geometry and uncertainty in Visual representations. Second we have confirmed that the gaussian assumption which allows us to use the powerful tool of Extended Kalman Filtering is quite adequate for both synthetic and real data obtained by a mobile robot operating in a human made environment Third, we have shown that these simple ideas provide an efficient mechanism for building up incrementally a coherent 3D representation of this environment while allowing the system to forget intelligently redundant information and, at the same time, improving the accuracy of its current estimation.

REFERENCES

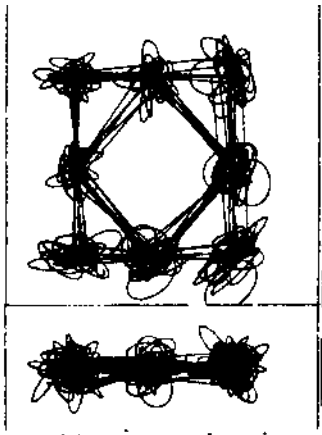
1 Ayache, N., and O.D. Faugeras, "Building, Registrating, and Fusing Noisy Visual Mips", ICCV'87, London, June 1987. (see included i)

Ayache, N., and OX). Faugeras, "Updating the Environment of a Mobile Robot", Proceedings of 4th ISRR, Santa Cruz, U.S.A.. August 1987.

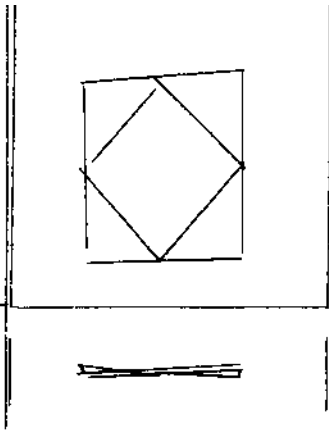
Ayache, N., and B. Faverjon, "Efficient Registration of Stereo Images by Matching Graph Descriptions of Edge Segments", IJCV, Vol. 1, N* 2, April 1987.

Ayache, N., and F. Lustman, "Trinocular Stereovision", in Proc. UCATO, Milano, August 1987.

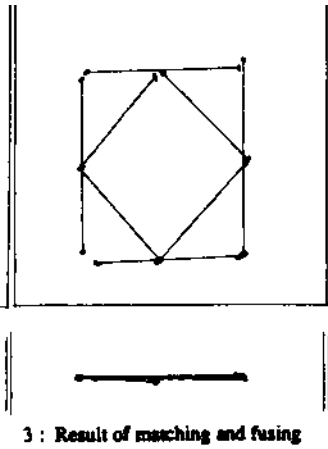
Faugeras, OD., N. Ayache, and B. Faverjon, "Building Visual Maps by Combining Noisy Stereo Measurements", Proceedings 1986 IEEE Conference on Robotics and Automation, San Fransisco, U.S.A., April 7-10, 1986, pp. 1433-1438.



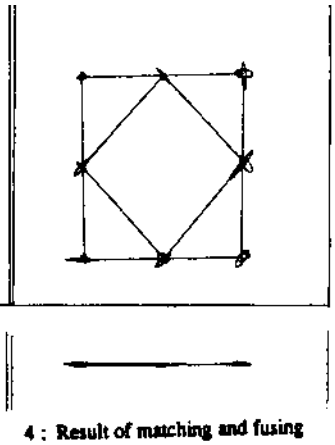
1 : Noisy segments and covariance matrices of the endpoints.



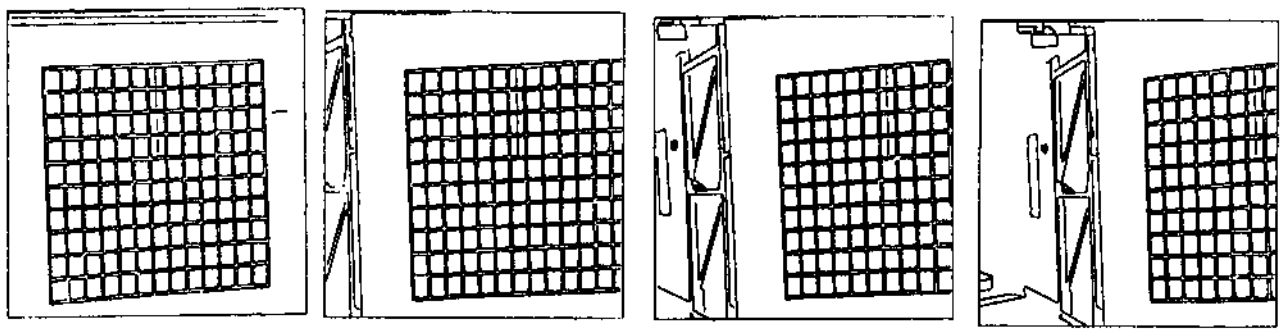
2 : Result of fusing using least squares : m.s. error = 4.3 %.



3 : Result of matching and fusing using Kalman Filter on the endpoints m.s.error = 3.2 %.

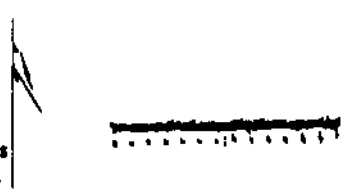


4 : Result of matching and fusing using Kalman Filter on the lines m.s.error = 1.15 %.



5 - 8 : The grid is observed by the stereovision system from 4 different positions.

9 : Vertical view of the reconstructed 3D segments placed into a common coordinate frame by the program.



10 : Automatic fusion of the previous 1808 segments into 650 segments.