T. Viéville, F. Romann, B. Hotz, H. Mathieu, M. Buffa, L. Robert, P. Facao, O. Faugeras, and J. Audren. Autonomous navigation of a mobile robot using inertial and visual cues. In M. Kikode, T. Sato, and K. Tatsuno, editors, Intelligent Robots and Systems, *Yokohama, 1993.*

Autonomous navigation of a mobile robot using inertial and visual cues

T. Viéville[1] and F. Romann[2] and B. Hotz[1] and H. Mathieu[1] and M. Buffa[1] and L. Robert[1] and P.E.D.S. Facao[1]
and O.D. Faugeras[1] and J.T. Audren[2]
(1): I.N.R.I.A., Rt des Lucioles, Sophia,06560 Valbonne, France
(2): SFIM, av M.R. Garnier, 91344 Massy Cedex, France

Abstract

This paper describes the development and implementation of a reactive visual module utilized on an autonomous mobile robot to automatically correct its trajectory. We use a multisensorial mechanism based on inertial and visual cues. We report here only on the implementation and the experimentation of this module, whereas the main theoretical aspects have been developed elsewhere.

Introduction

This study aims at developing a method of recovery of the ego-motion and the 3D-structure of a scene, in the case of a virtual moving *observer* with visual, inertial and odometric sensors. This observer attempts to correct its trajectory and build a 3D depth map of its environment during the execution of a predefined trajectory (a prerecorded list of relative displacements).

More precisely, we are going to report on the development and implementation of a reactive visual module utilized on an autonomous mobile robot to automatically correct the predefined trajectory, using inertial and visual cues. We describe only the implementation and the experimentation of this module, whereas the main theoretical aspects have been developed elsewhere [7, 21, 9].

This work must be considered as a *new method* but as an experimental report only. A discussion on ego-motion estimation of the mobile robot can be found in [24] for instance, while several experiments on vision-based autonomeous navigation have been already published by many authors (see for instance [5, 3, 10, 14, 8, 12]).

We just have attempted to add a few set of experimental results with respect to what the scientific community has already acquired in the field. Precisely, we have tried to perform these tests considering the following paradigm: (1) a 3D dense stereo system, (2) a low-cost inertial system coupled to feedback a mobile robot on its initial trajectory when errors occur. In indoors and outdoors environment. In real-time.

Choosing a method of stereo-vision.

Over the years, we have developed numerous algorithms for passive stereo [2, 20, 16, 7, 19, 18]. The first class of algorithms extract features of interest from the images, such as edge segments, curves or chained contours, and match them in two or three views [2, 20]. Such methods yield very sparse depth maps, only located on contours points. Although they yield a symbolic representation of the reconstructed scene, their use is limited to the class of scenes where edges are relevant features to analyse.

The second class of algorithms is based on areas correlation [7, 9] or photogrammetric methods [18] and produces denser maps with very few false matches. The reconstructed scene is, in this case, not a list of structured data but a depth map, more precisely a 3D map of the visual surroundings in front of the robot.

Moreover, we want to consider the case of actual available sequences of time-varying images sampled at 0.5 to 30 Hz. In such paradigms the computation time between two consecutive views must be very small, and the related equations on motion and structure must be computed in real-time [6, 20, 7].

In this paper we are going to describe a real-time stereo correlation algorithm (a large part of the following description is extracted from [7]). Then we analyse the behavior of the algorithm for autonomous navigation.

Parameters are defined using a linearized statistical representation : estimations are related to the minimization of pseudo *Mahalanobis* distances and controlled using *chi-square* tests, as usually done in the field.

Using inertial cues for ego-motion estimate

The use of inertial measurements on a robot has been already studied in previous papers such as [22, 21]. In particular, it has been possible to implement and calibrate a low cost inertial system on a robot, in order to obtain information about the robot ego-motion [23]. Inertial measurements yield the same kind of information as those provided by passive navigation algorithms using artificial vision, but with a different dynamic range and precision. Thus, cooperation between these two sensory modalities may be useful for the elaboration of high-level representations.

On earth, there exists a constant homogeneous gravity field always and everywhere present, defining an absolute 3D-direction, and this 3D-direction can be measured either from inertial sensors [22], or by the determination of vertical lines in an image (see [13, 21] for instance). We must thus introduce 3D-cues at an early stage of the process.

On a robot, two types of inertial informations can be computed : The instantaneous **ego-motion** of the robot (also called either self motion or vection), and the **vertical angular orientation** of the robot in space.

The available inertial sensors, for a robotic visual system, are linear accelerometers and either angular rate sensors or gyroscope. Their costs are similar to the cost of a visual sensor.

Angular rate sensors provide an information about angular velocity with a resolution better than 0.04 deg/sec for the low cost units, their overall precision being similar to the linear accelerometers. After calibration [22] they provide an estimate of the angular velocity ω, though a relation of the form :

$$w = \omega + \mathcal{N}oise \tag{1}$$

Integrating this equation as in [22] yields a local estimate of the robot orientation. Another way is to use a gyroscope.

We have a short term estimate of the horizontal and vertical orientations from this sensor, thus a short term estimate of the vertical.

Low cost linear accelerometers have a precision of about 0.5 % with a scale of measurement between $\pm 2 \times 9.81 m/s^2$, while the resolution is of 0.1 % in a $0 - 50\ Hz$ range. After calibration

[22] they provide an estimate of the specific forces $\mathbf{a}$:

$$\mathbf{a} = \gamma + \mathbf{g} + \mathcal{N}oise \tag{2}$$

where $\mathbf{g}$ denotes the vertical gravity field, and $\gamma = \ddot{M}$ the true acceleration.

Any robotic system, on earth, is embedded in the gravity field, and it has been shown that the measurement of the absolute vertical is possible using low-cost inertial sensors [23]. The key point is that the gravity acceleration is constant, homogeneous, and defines the absolute vertical axis. It thus constitutes a basic cue for spatial orientation. We can, using this, relate our visual informations to an absolute frame of reference, with the vertical as a fixed axis[1].

Moreover, it is possible to compute the vertical direction in a natural visual scene, since it often contains vertical lines which intersect at infinity in 3D and a vanishing point in the image, which can be detected (see for example [21]). In addition, on a mobile robot, an approximate orientation of the camera is known, and this allows to approximate the location of the vertical direction.

What is the paper about

In the first section we discuss the implementation of short-term trajectory corrections on an autonomous robot, using inertial cues.

The second section is a presentation of the real-time stereovision method implemented for the visual correction of the predefined trajectory.

In the third one, we report experimental results on the visual correction of the robot trajectory and demonstrate the robustness and the efficiency of the method in indoors and outdoors environments.

1 Using inertial cues to correct the robot heading

In the case of our application, the inertial system allows to compute the horizontal orientation (heading) with high accuracy considering short periods of time. It returns this information as a regularly updated value, but without any synchronization with respect to the odometric measurements. We have experimented a square trajectory and pushed randomly the robot to introduce an unexpected error for the heading. To perturb also the odometric correction, the perturbation is done when the mobile robot is on the skating (or slippering) floor. If no correction from the inertial system occurs, the robot ends its trajectory with a large rotation error.

1.1 Control of the robot trajectory with inertial cues

In order to describe the control we have used in our implementation we must analyse the kinematic of the mobile robot. Considering the diagram of Fig. 1, and knowing that the trajectory of such a mechanism is always locally circular (sometimes rectilinear if the radius is infinite) we can easily obtain the kinematic equations of such a system.

[1] It is not possible *a priori* to separate gravity and true accelerations, but it has been suggested [23] that two reasonable hypotheses can be used to separate those two components : (1) $\mathbf{g}$ is a constant uniform 3-D vector field, (2) it is not physically possible to keep acceleration constant. The former hypothesis is formalized by : $\dot{\mathbf{g}} = \omega \wedge \mathbf{g}$ while the latter is related to the signal spectrum.

With these hypotheses [23] one can separate the two components of the specific forces, and we obtain $\mathbf{g}$ and γ from two differential equations, with initial integral conditions :

$$\begin{array}{rclrcl} \dot{\gamma} & = & \dot{\mathbf{a}} + \omega \wedge (\gamma - \mathbf{a}) & \int_{-\infty}^{0} \gamma(t)dt & = & 0 \\ \dot{\mathbf{g}} & = & \omega \wedge \mathbf{g} & \int_{-\infty}^{0} \mathbf{g}(t)dt & = & \int_{-\infty}^{0} \mathbf{a}(t)dt \end{array} \tag{3}$$

We thus have a long term estimate of the vertical orientation from this sensor.

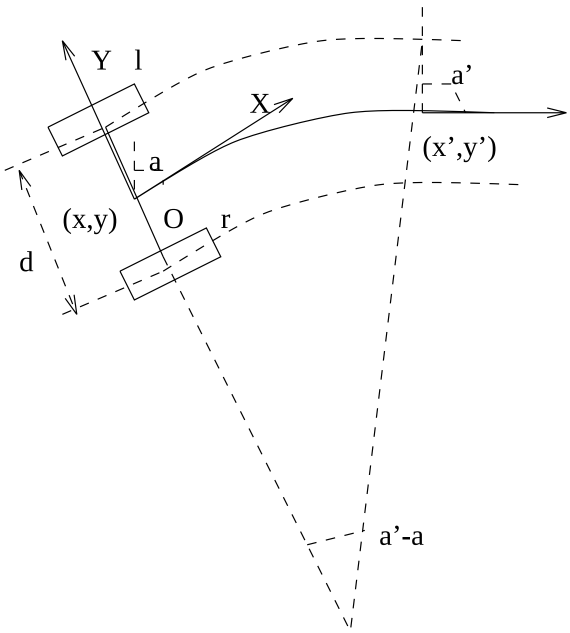

Figure 1: Kinematic of the mobile robot, with differential wheels.

The mechanical parameters are the diameter D of each wheel, and the distance d between the two wheels. The first quantity is well defined for a rigid wheel but subject to variations for a wheel with a tyre. Our experience is that this parameter is quite stable and that variations cancel on average in time (the long term precision is better than $0.001 = \frac{1\,cm}{10\,m}$). However, the distance between the two wheels is a much less precise quantity, because it can change with the trajectory, for instance during a left or a right turn. This is due to the fact that this distance is a function of the nature of the contact between the robot wheel and the floor, thus not measurable. Let us analyse the consequences of this problem and demonstrate that our inertial measurement is well adapted to solve it.

The curvilinear displacement l and r for the left and right wheels are very easy to relate to the wheel diameter d and the angular displacement of the (say the left one) wheel t since $l = \frac{D}{2}\,t$. The current Cartesian position of the robot (x, y) and its orientation a can be related to the next Cartesian and angular positions using the equations :

$$\left\{ \begin{array}{lcllcl} a' & = & a + \frac{r-l}{d} & & & \\ x' & = & x + \frac{r+l}{2}\frac{\sin(a'-a)}{(a'-a)} & = & x + \frac{r+l}{2} & \left[1 - \frac{(a'-a)^2}{3} + o((a'-a)^4)\right] \\ y' & = & y + \frac{r+l}{2}\frac{(\cos(a'-a)-1)}{(a'-a)} & = & y + \frac{r+l}{2} & \left[\frac{(a'-a)}{2} + o((a'-a)^3)\right] \end{array} \right.$$

which are easy to obtain, considering a frame of reference attached to the robot.

The analysis of these equations yield several properties :

- The lateral $(y' - y)$ and longitudinal $(x' - x)$ displacements are not directly dependent of d but only on the angular displacement of the robot $(a' - a)$.
- The angular displacement of the robot $(a' - a)$, as given by the measure on the wheels is a function of the unstable parameter d
- An error on the angular displacement of the robot has a direct linear influence on the robot lateral displacement, whereas its influence on the longitudinal displacement is of the second order. Moreover these errors accumulate with time.

These remarks confirm that the crucial parameter to correct, considering odometric cues, is the angular position of the robot; the lateral displacement of the robot is subject to a cumulative error whose amplitude is a linear function of time and of the angular position error.

As a consequence the simple strategy which is to combine inertial and odometric cues to estimate the angular displacement of the robot is very suitable and has the effect to correct the main errors which occur during the execution of a predefined trajectory.

The residual lateral correction can also be obtained by integrating the angular orientation provided by the inertial system with the linear displacement of robot measured using the odometry. However the integration of a linear position from inertial cues is known to be subject to errors quadratic with time, and in our configuration, the odometric cues are more efficient. In addition we have developed a better mechanism to obtain this lateral correction : the use of visual cues.

Finally, a small longitudinal error is not essential to correct for road navigation, for two reasons : (1) a small error in the direction of the robot heading is not going to put the robot out of the road as a lateral error would, especially for rectilinear parts of the road, but (2) for curved parts of the road (considering a 90 *deg* turn for instance) this error is in fact reported as a lateral error, thus going to be corrected by the system. Remember this error is quite small is any case.

1.2 A simple adapative correction using the inertial cues.

The error is calculated as the difference between the expected and the measured headings. In order to take into account the fact that we do not know the exact time when the measure was made (sometime between the present and last sampling times), this angular value is estimated as the mean between the two intermediate values. The heading is corrected for the next displacement, and the heading error is calculated when a new command is sent to the mobile robot. The resulting equations of heading correction are:

$$\begin{aligned} ErrorHeading &= \frac{LastInertialHeading + PresentInertialHeading}{2} \\ &- ExpectedHeading \\ NextHeading &= ExpectedHeading - \kappa ErrorHeading \end{aligned} \tag{4}$$

This result is shown in Fig. 2.

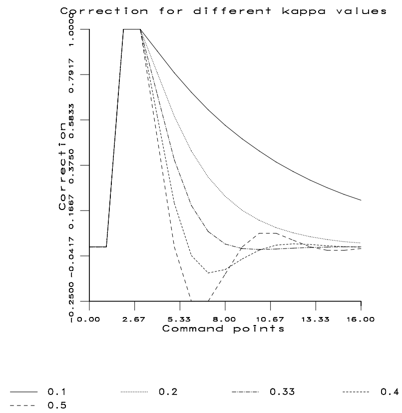

Figure 2: The corrections using several values of κ, small values of κ.

When we analys this graphic, we notice that the correction can't be effective if κ is greater than 0.33. In fact, it is easy to verify that the gain of the correction must be 1/3 in order the correction to asymptotically stable[2] Experimentally, the more κ value increases the more the robot oscillates. If κ=1 the error can't be corrected.

The position is partially corrected but not entirely because we do not know exactly when the perturbation of the robot displacement has occurred. The original trajectory is shifted (lateral error) but the robot closes the square without any angular error.

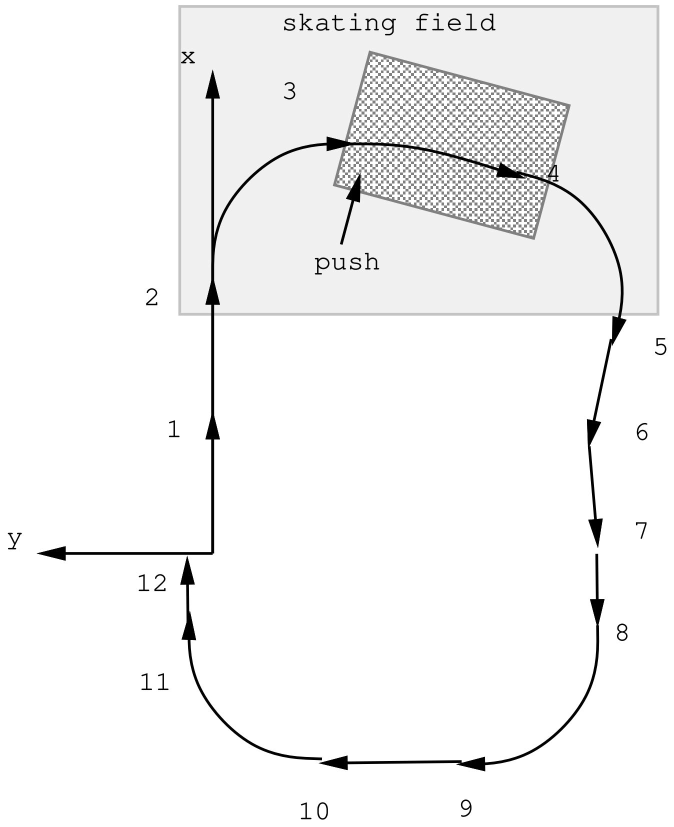

Figure 3: Correction of the perturbed trajectory.

Another technic would have been to use prediction in order to compensate for the delay introduced in the system. We have experimented such mechanisms but the results are not convincing. This is due to the fact that the synchronization between inertial measurements and odometric cues is not very precise in our configuration. Therefore the prediction of the robot orientation is quit uncertain and thus yields unstabilities.

Moreover the robot is subject to local perturbations related to the feedback mechanisms. These perturbations are unpredictable and induce wrong values of the Heading but they are quickly corrected by the odometric system, and thus must not be taken into account by the inertial correction module.

Therefore, this simple correction seems to be a nice compromise for the control of a mobile robot.

We have now to describe how the vision can compensate for lateral errors. Let us first

[2]Let us consider a step-like error detected at time $t0$. When this error occurs the system is already calculating the next trajectory. At time $t0+1$ the new command is sent and the error is determined. At time $t0+2$ the trajectory is not corrected but it detects the same error again and sends the corrected order to reach the next point. At time $t0+3$ the error is null but the system sends the new command with a correction proportional to the error detected on the previous point. It is the consequence of the delay between the error detection and its correction, that the error is taken three times into account.

describe the stereo vision algorithm and then report on methods used to compute the lateral correction of the robot.

2 The stereo correlation algorithm

We have implemented and improved the original method developed by [7, 9]. Since we have experimented and reimplemented this method in the scope of this study, a precise description is given in appendix 5.

In this section we describe how the standard algorithm can be used, the parameters set, and on a very large set of images[3] explain in detail the behavior of the algorithm.

2.1 Standard algorithm and parameters

The parameters of our stereo-correlation algorithm are:

- the similarity criterion (C_2, C_5 or C_6 as previously defined)
- the correlation window size: (ni, nj)
- the range of disparities examined: $[dmin, dmax]$
- the number of successive erosions and dilatations in the elimination of isolated matched points: $elim$
- the number of resolution levels: $level$

We could have chosen a normalized correlation score to avoid problems of different camera sensitivities. The C_5 criterion seems to give a slightly better results than C_6, and thus can be used.

According to the fact that disparity values can be very different from one stereo pair to another, we have decided to adjust for each pair a wide enough interval of disparity so that all good matches can be found. It is equivalent to fix a depth zone in the scene. In fact, the interval of disparity can be automatically computed if the relationship between disparity and depth is known. Moreover if we spread the interval of disparity, the results in most cases are not modified in a significant way, except on repetitive patterns.

The medium value of 3 is used for the $elim$ parameter which monitors the size of isolated groups of matched points to be eliminated. At last we have decided to use only the finest level of resolution for our standard algorithm to have precise results.

In fact, for the real-time implementation the following restrictions have been made : (1) we have limited our implementation to the criterion C_2 which is much more efficient to compute, (2) the erosion-dilatation mechanism is not yet implemented, and (3) neither the hierarchical resolution nor the computation of precision maps are computed. But on the workstation various possibilities have been experimented in order to have a precise idea of the behavior of the program.

We have tested about ten representative stereo pairs to determine the correlation window size and have found that a 9×9 window is a good compromise between matched point density and disparity value accuracy on the whole data set, for 128×128 images.

The output of the correlation program is an image of disparities, an image of confidences and an image of precisions as explained before. Moreover information on the qualitative behavior of the algorithm for a given point can be obtained[4]. Finally the depth map of the visual surroundings can be computed.

---

[3] It has been experimented on the whole JISCT data set provided by the SRI, thanks to B.Bolles.

[4] We can distinguish the following annotations for each pixel of the disparity map :

- case 0: No match attempted on the edge of images (due to the fact that the correlation window cannot go out of the images)
- case 1: Match is fine
- case 2: No match the correlation curve is too flat

Two examples of stereo reconstructions are shown in Fig. 5 for the view of Fig. 4 corresponding to an indoor scene, and in Fig. 5 for the view of Fig 4 corresponding to an outdoor scene.

Figure 4: An indoor scene used for the stereo

2.2 Overall analysis of the algorithm

Let us report on the behavior of the algorithm, using an image data base of various images (45 images).

General behavior

- It produces very dense maps (works on each point of the images).
- It yields reliable (instead of making mistakes, the algorithm produces no answer) and precise results (subpixel evaluation of disparities by locally fitting a parabolic curve to the correlation peak).
- It can easily detect and eliminate gross errors (errors are mostly situated in sparse regions of the disparity map, except for repetitive patterns where they have very low confidences).
- It can produce more information about matches with a small increase in computation time: annotations (not yet implemented), confidences, precisions...
- It can use several levels of hierarchy to make the disparity map denser.

- case 3: No match because of low match value when the correlation score is below a fixed threshold.
- case 4: No match because of multiple choices (repeated structure) when the confidence value is very low.
- case 5: No match because of uncompatibility of the disparities in the two directions (our validity criterion).
- case 7: No match because of occluded areas (more difficult to detect, examine the neighbourhood).

Figure 5: The depth map of the 3D reconstruction for the indoor scene, light values correspond to important depth, dark values to small depths.

- It has a small number of parameters which can be easily determined for a known type of scene (the quality of the results does not depend heavily on the value of these parameters, in particular on the window size and the number of hierarchical levels).

Implementation considerations

- It is a very simple algorithm, no complex criterion is used to match points.
- Its is a very regular algorithm, which can be easily implemented on massively parallel architectures like the Connection Machine or on special hardware (we have implemented this correlation algorithm on a four Motorola DSP96002 board and have obtained a processing time of less than 1 seconds for a 128×128 images). This includes the picture rectification (the operation which allow the epipolar lines to be parallel) and the 3D reconstruction.
- Its complexity is only proportional to the number of pixels in the images and to the number of disparity values examined in the matching process, in particular, the algorithm is implemented so that the processing time does not depend on the size of the correlation window.

Detailed behavior

- Small or thin obstacles may not be detected if the window size is too large (as observed on trees).
- It produces gross and dense errors on repetitive patterns which sometimes cannot be eliminated with successive erosions and dilatations (such problem can be detected by examining the form of correlation curve and the confidence value; horizontal repeated patches can be correctly matched if a third upper camera is used) (as observed on synthetical images).

Figure 6: An outdoor scene used for the stereo

- Moving objects can be matched if their appearance has not changed a lot and if the interval of disparity is wide enough (as observed when looking at a walker).
- It is relatively little sensitive to dynamic range of intensities in spite of the fact that criterion depends directly on intensity values but is insensitive to gray level difference between stereo images due to the use of a normalized criterion (C_5) in the matching process.
- It is sensitive to noise when using small windows (unstable correlation curve; increase the window size makes the correlation curve smoother).
- No answers are found on occluded areas, except with large windows were occluded areas are filled by wider objects.
- Bland areas often cause sparse maps, but sometimes dense errors are generated for zones of constant intensity values (a test of the form of correlation curve can be added).
- It fails (no answer) for strong texture gradients (problem resolved using hierarchical algorithm to perform the correlation using several frequency bands of the image signal).
- It produces no matches on horizontal features (resolved using a third camera up or above the two others and performing a correlation in the vertical direction).
- Taking a larger interval of disparity does not modify the results in a significant way (our validity test is robust), except on repetitive patterns.
- Objects visible only in one image are of course not correlated.
- Imprecisions in the disparity values are strong when the correlation peak is too flat (caused by the use of large windows, coarse levels of resolution, bland areas...).
- The use of large correlation windows (or hierarchy) yields less precise of objects locations (the condition of constant disparities all over the window is violated in the case of depth discontinuities like on object edges, and produces sometimes fatter objects).

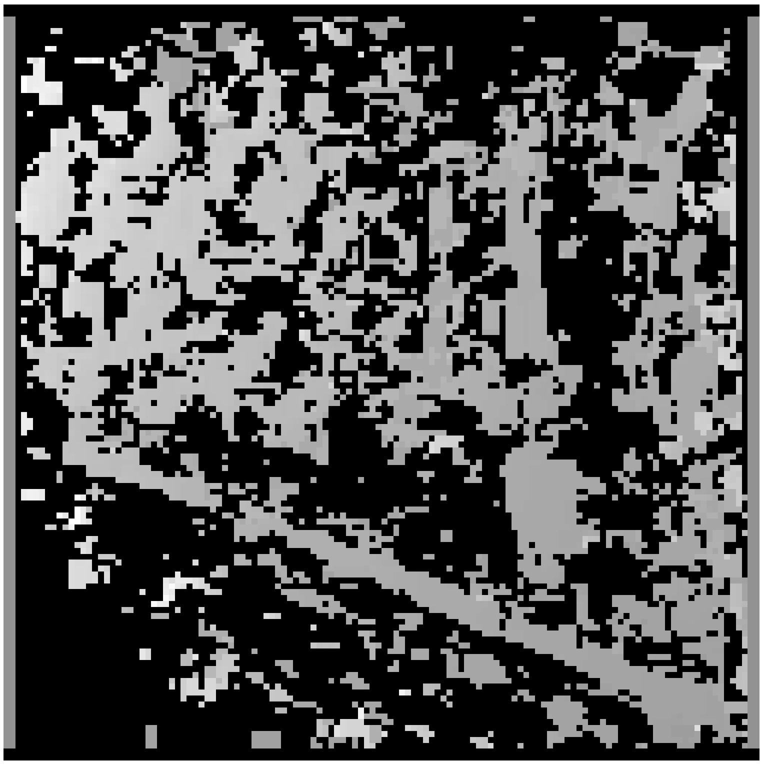

Figure 7: The depth map of the 3D reconstruction for the outdoor scene, light values correspond to important depth, dark values to small depths.

- It fails (no answer) when epipolar lines are not totally horizontal or when there is a gap between them, such that y disparities cannot be neglected (our algorithm makes the assumption that images are precisely calibrated; a method to avoid this problem is to search the corresponding points on a group of several lines rather than only one line).
- Subsampling the lines of the images does not degrade correlation results in a significant way.
- When the baseline is too large, the correlation algorithm produces very sparse maps with a few errors.

3 Introducing 3D vision to correct the robot trajectory

In a the first section we have described how the angular orientation of the robot can be corrected locally using inertial cues. The problem has not been solved completely, since a lateral deviation of robot position still has to be corrected.

3.1 Computing angular and lateral error using 3D vision

Now, having a 3D map of the scene it is very easy to detect the road/corridor since it corresponds to a flat region of the scene "at height 0", in front of the robot. We have detected edges of the road/corridor by thresholding the height of the 3D scene and have extracted the left and right limits of the road/corridor (which have been modeled as locally rectilinear edges).

First considering these two parallel lines one can compute their median line, the axis of the road. Second we know from calibration the axis of the robot. The lateral displacement can thus be defined as the distance between the road axis and the robot axis at the robot location, the error being not zero if these two lines are not the same.

Moreover, detection of outliers measures of lateral errors is easily implementable. We reject values which are too important, and accept the value if and only if, considering an approximate value of the robot width and the road size, this correction cannot put the robot on the road size.

This is shown in Fig 8. The angular error can also be computed with this representation and used to reset inertial measurements.

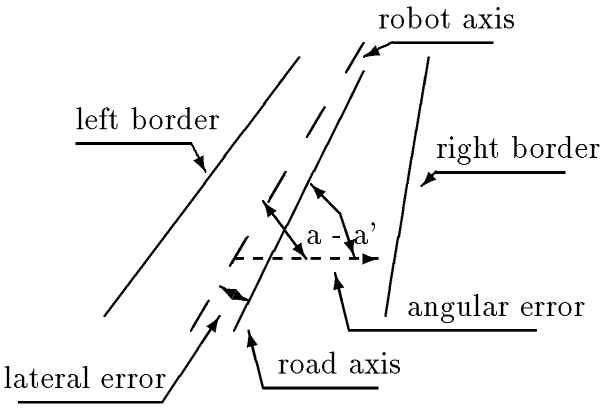

Figure 8: The definition of the lateral and angular error of the robot on the road/corridor

The general algorithm has been implemented as follow :

- Split the depth map into two parts corresponding to the left and right parts of the road, using the initial estimate of the road axis obtained from the positioning of the robot.
- For each (left and right) part :
  - Define a few set search areas for the road size which are located at different distances in front of the robot :
  - For each area :
    - Compute the histogram of the height of the road.
    - Threshold and filter the histogram.
    - Search for the first points at a non-zero height, they yield location of the side of the road.
  - Compute the mean and variance of the edge location for the set of measures obtained in each area
- Compute the angular and lateral error of the mobile robot (mean and covariance matrix).
- Fuse the information with the inertial and odometric cues (Kalman filtering).

Note that we have exploited the simplicity of the scenes to avoid the use of very heavy "every cases computations". When defining the trajectory in terms of a list of absolute positions/orientations to attain, we have also recorded the where visual information can be reliably computed and the areas where it cannot (because we expect nothing to see for instance). Moreover, we have a very simple thresholding mechanism to reject unreliable estimate of the road edges (for instance the road width must be in a certain interval, the edges relative orientation must not differ too much, etc...). We have observed that these heuristics allow to reject easily erroneous estimates, since either there is something to see, and the estimation is reliable, or there is nothing to see and the estimate is rubbish. As a consequence the adjustment of these parameters is not very crucial, and not very sensitive.

3.2 Experimental results

We have obtained a robust autonomous navigation for several minutes with a mobile robot running at 1m/sec indoors (the corridor width was 1.6 m while the robot width was 0.8 m) and 1.5m/sec outdoors (the road width was 5 m while the robot width was (still !) 0.8 m) the limit being the motor torque, not the algorithm.

Analysis of the visual process For indoors scenes the algorithm provides relevant values mainly on edges and is thus not qualitatively much more efficient than feature based stereovision methods; useful parts of the scene are : doors/windows, limits of dividing walls, posters or painted lines on the floor or on the walls. But for outdoors scenes, the system can integrate much more informations, since useful parts of the scene are : shadows or plates on the road, vegetation, parked cars, sidewalk; in that case, this dense stereoscopic method is preferable.

The computation time is less than a second for the computation of the 3D map (128×128) which is performed on the DSP board MD96 and 0.35 sec for the computation of the road edges on a Sun4-II Sparc station.

In principle, the technique for the road extraction is sensitive to the number and quality of the points detected as belonging to the road edge. We thus have measured the stability and the repetability of the method for several set of measures. As in the algorithm, the estimation has been made taking into account the variance of each estimate and thus using an optimal estimation in the least-squares sense (maximum likehood, which reduces in that case to the minimization of a criterion weighted by the variances). We have obtained the following three results :

	Exp. 1	Exp. 2	Exp. 3
Expected value (mm)	300	-300	200
Number of measures	13	16	12
Optimal mean value (mm)	301	-303	212
Optimal standard deviation (mm)	83	68	71

In fact, we have observed that when the measure are not very stable as in Experiment 1 which has been made in a degraded situation, the integration of the variances in the estimate yields much better results (the simple mean and standard-deviation where : 231 ± 215). It demonstrates that the estimated variances are meaningful and must be taken into account.

Analysis of the compensation In order to experiment our system we had to artificially increase the navigation errors because the system was much too stable to observe a relevant correction.

We thus have introduced a drift in the inertial system of 50 to 200 deg/hour. We also have voluntarily introduced some errors in the trajectory : (1) an initial lateral and longitudinal error from 0 to 20 cm and up to 15 deg in orientation, (2) a continuous error (lateral translation) along the trajectory up to 10 cm for each sampling period, (3) some step-like errors up to 50 cm lateral and longitudinal. For the previous quantities the mobile robot was always capable to recover, except in case of a transmission problem[5]. For higher errors, we may not be able to correct the error, and this gives an idea of the limits of our implementation at the present stage.

In order to insure the stability of the mobile robot guidance, we have implemented a compensation mechanism which can take into account inertial and visual cues to compensate both angular and lateral errors, but with a weight in favor of the inertial cue for the angular error, and the visual cue for the lateral error. With this simple strategy (obtained by a correct balance of the variances in a standard Kalman filter) we have obtained a stable control, we have observed the capability of the inertial system to act as a short-term correction, especially when the vision

[5]We definitely cannot claim that the our robotic system has a robustness which would allow too use it in an industrial environment as it is. This is only a "lab. demo.". But in any case, the limits we have encountered were only due to the fragility of such an experimentation, not to the method. Suspicious readers must know that, despite this restriction, the mobile robot is still alive.

was not usable, and have verified that the vision can perform a long-term correction and thus cancel the inertial system drift.

At the beginning of a trajectory (bootstrapping phase), the system needs 1 to 3 sampling periods to obtain a correct visual correction, and thus the initial velocity must be rather small (10 cm/sec indoors and 30 cm/sec outdoors). In the steady-state phase, errors to be corrected are less than 1deg and 10cm indoors and less than 1deg (2deg during the first minute) and 50cm outdoors. In agreement with these results the uncertainty (standard deviation) of the measures initialized at 5deg and 50cm decreases quickly to 0.5deg and 10cm indoors and 0.5 deg (2deg during the first minute) and 30cm outdoors.

A few measures are much less precise (1.5m of standard deviation) and thus not taken into account. In detail, outdoors, more than 80% of the measures have less than 30cm of uncertainty, whereas only 5% have more than 50cm of uncertainty.

A few set of recorded data We show some results indoors in Fig. 12 and Fig. 13. With more details, because this is a more difficult task, considering the nature of the scene, we have shown two examples of corrections Fig. 9 and Fig. 11. The positioning of the robot after its correction obtained from the measure shown in Fig. 9 is shown in Fig. 10.

Lines have been drawn in the pictures to allow the analysis of the behavior of the visual compensation :

- A white line corresponds to the axis of the road at a distance of 10m, in the estimated position before the visual compensation. It is drawn as if on the floor.
- Three black lines corresponds to the left/right edges and the axis of the road, as found in the 3D reconstruction; they intersect.
- Two light polylines join the points which have been detected as road edges, for the left and right edges.
- The white line and the three dark lines are also drawn in a top view in the upper part of the image, without any relation with the image itself.

With this representation the reader can easily observe the good results of the algorithm. The image sequences contain not only the "best" results but also some estimations which have been rejected by the algorithm as described previously. This case is easy to detect on the picture considering the road estimations reported in the images.

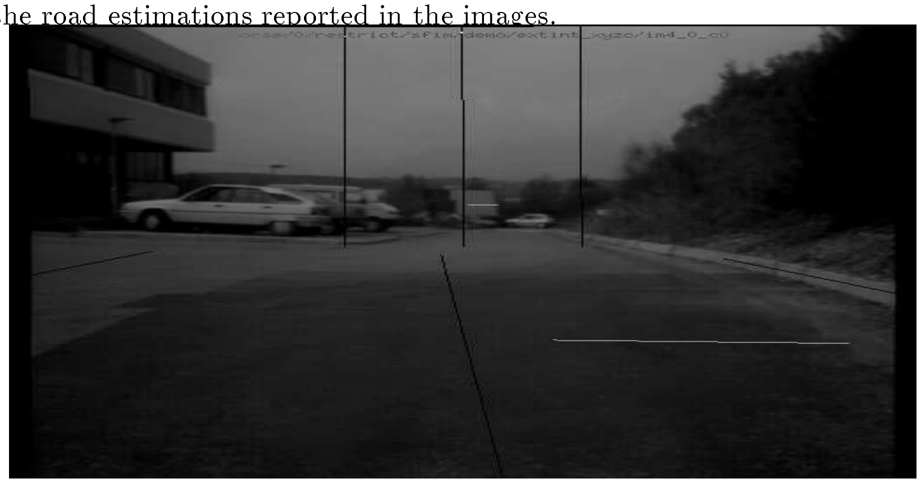

Figure 9: One example of lateral correction, outdoors, see text.

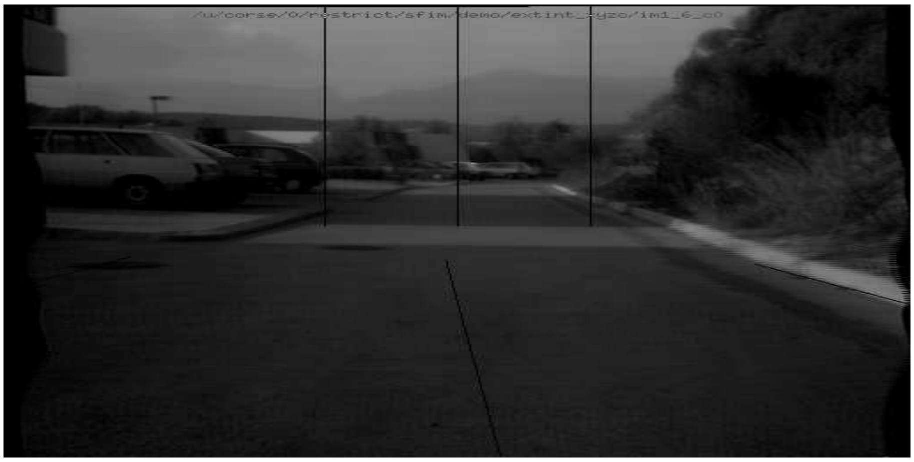

Figure 10: The positioning of the robot after the previous correction, see text.

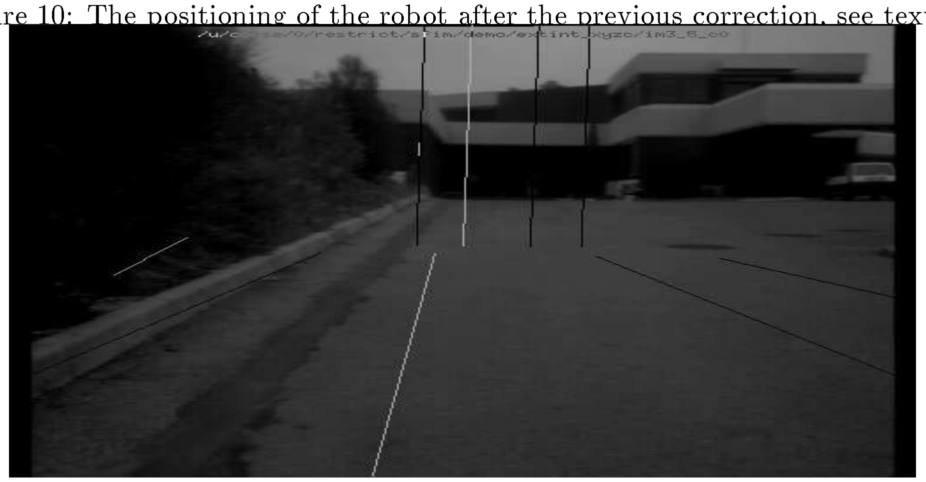

Figure 11: Another example of lateral correction, outdoors, see text.

4 Conclusion

This mechanism is an example of an operational and efficient use of reactive vision, with multi-sensor fusion of inertial and visual cues. Using this mechanism, we can correct for positioning errors and recenter the mobile robot on its trajectory. It is quite sophisticated at the implementation level, but its architecture is a simple instanciation of the following rule for the multi-sensor cooperation :

- Inertial cues correct the robot trajectory for short term errors, and mainly its orientation.
- Visual cues, in this case, correct long term errors and mainly the lateral bias of the robot on its trajectory.

A 3D model of the visual surroundings is generated by the system and can be used for higher level mechanisms of perception.

The system is "cheap" in the following sense : we need only two cameras, an image acquisition module, a one axis low-cost gyroscope and a 4-DSP high-speed computer board[15] for a rate of correction of 1.5 sec (computation time).

References

[1] P. Anandan. A computational framework and an algorithm for the measurement of motion. *International Journal of Computer Vision*, 2(3):283–310, 1989.

Figure 12: A session of autonomous navigation, indoors, see text.

[2] N. Ayache. *Artificial Vision for Mobile Robots.* MIT Press, Cambridge, Massachusetts, 1989.

[3] B. Bhanu, P. Symosek, S. Snyder, B. Roberts, and S. Das. Integrated binocular and motion stereo in an inertail navigation sensor-based mobile vehicle. In *I.E.E.E. Conf. on Intelligent Control, Glasgow*, 1992.

[4] P. Burt, C. Yen, and X. Xu. Local correlation measures for motion analysis. In *IEEE PRIP Conference*, pages 269–274, 1982.

[5] S. Cornell, J. Porill, and J. Mayhew. Ground plane obstacle detection under variable camera using a predictive streo matcher. Technical Report AIVRU 73, University of Sheffield, 1992.

[6] O. D. Faugeras, R. Deriche, N. Ayache, F. Lustman, and E. Giuliano. Depth and motion analysis: the machine being developed within esprit project 940. In *Proceedings of the IAPR Workshop on Computer Vision (Special Hardware and Industrial Applications), Tokyo, Japan*, pages 35–44. Institute of Industrial Science, University of Tokyo, October 1988.

[7] P. Fua. A parallel stereo algorithm that produces dense depth maps and preserve image features. *Machine Vision and Applications*, 1991.

[8] M. Hebert and T. Kanade. 3d vision for an autonomous vehicle. *I.E.E.E. Trans. on Robotics and Automation*, 3:375–380, 1987.

[9] B. Hotz. Etude de techniques de stereovision par correlation . application au programme vehicule autonome planetaire (v.a.p.). Rapport de stage de DEA TE/AE/SE/SR No 91/242, Centre National d'Etudes Spatiales, Toulouse, Septembre 1991.

[10] R. Jarvis. An autonomous mobile robot in a rangepic world. In *2nd Conf. on Automation Robotics and Computed Vision, Singapore*, 1992.

[11] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptative window: Theory and experiment. In *Image Understanding Workshop*, September 1990.

Figure 13: Another session of autonomous navigation, indoors, see text.

[12] O. Khatib. Real-time obstacle avoidnace for manipulators and mobile robots. *Int. J. Rob. Res.*, 5:90–98, 1986.

[13] X. Lebègue and J. Aggarwal. Detecting 3D parallel lines for perceptual organization. In *2nd ECCV*, pages 720–724, Genoa, 1992. Springer Verlag, Berlin Heidelberg New-York.

[14] J. Leonard and H. Durrant-Whyte. Mobile robot localization by tracking geometric constraints. In *I.E.E.E. Conf. on Decision and Control*, 1990.

[15] H. Mathieu. *Multi-DSP 96002 board technical manual*. INRIA, Octobre 1991.

[16] N. Navab, R. Deriche, and O. Faugeras. Recovering 3-D motion and structure from stereo and 2d token tracking. In *Proceedings of the 3rd ICCV, Osaka*, 1990.

[17] H. Nishihara and T.Poggio. Stereo vision for robotics. In *ISRR83 Conference*, Bretton Woods, New Hampshire, 1983.

[18] L. Robert, R. Deriche, and O. Faugeras. Dense Depth Recovery From Stereo Images. In *Proceedings of ECAI*, pages 821 – 823, Vienna, Austria, August 1992.

[19] L. Robert and O. Faugeras. Curve-Based Stereo: Figural Continuity and Curvature. In *Proceedings of CVPR*, pages 57–62, June 1991.

[20] R. Vaillant, R. Deriche, and O. D. Faugeras. 3-D Vision on the Parallel Machine CAPITAN. In *International Workshop on Industrial Application of Machine Intelligence and Vision*, april 1989.

[21] T. Viéville, P. Facao, and E. Clergue. Computation of ego-motion using the vertical cue. *Machine Vision and Applications*, 1994. To appear.

[22] T. Viéville and O. Faugeras. Computation of Inertial Information on a Robot. In Hirofumi Miura and Suguru Arimoto, editor, *Fifth International Symposium on Robotics Research*, pages 57–65. MIT-Press, 1989.

[23] T. Viéville and O. Faugeras. Cooperation of the inertial and visual systems. In T. Henderson, editor, *Traditional and Non-Traditional Robotic Sensors*, pages 339–350. Springer-Verlag, Berlin, Sept. 1989.

[24] C. Wang. Location estimation and uncertainty analysis for mobile robots. In *I.E.E.E. Conf. Robotics and Automation, Philadelphia*, pages 1230–1235, 1988.

5 Description of the stereo correlation algorithm

A number of correlation-based algorithms attempt to find points of interest on which to perform the correlation. This approach is justified when only limited computing resources are available, but with modern hardware architectures it becomes practical to perform the correlation over all image points and retain only matches that appear to be "valid". The hard problem is then to provide an effective definition of what we call validity and we will propose one below. We improved and parallelized the original method developed by [7].

5.1 Matching process

Correlation scores are computed by comparing a fixed window in the first image to a shifting window in the second. The second window is moved in the second image by integer increments along the corresponding epipolar line and a curve of correlation scores is generated for integer disparity values. The measured disparity can then be taken to be the one that provides the most important peak. To compute the disparity with subpixel accuracy, we fit a second degree curve to the correlation scores in the neighborhood of the extremum and compute the final disparity by interpolation.

5.2 Correlation criterion

To quantify the similarity between two correlation windows, we can choose among many different criteria the one that produces reliable results in a minimum computation time. We have tested the four criteria defined below (we take the case of horizontal epipolar lines on the same image line, so that we have no vertical (y axis) disparity):

$$C_1(x,y,d) = \frac{\sum_i \sum_j [I_1(x+i,y+j) - I_2(x+d+i,y+j)]^2}{\sqrt{\sum_i \sum_j I_1^2(x+i,y+j)} \times \sqrt{\sum_i \sum_j I_2^2(x+d+i,y+j)}}$$

$$C_2(x,y,d) = \frac{\sum_i \sum_j I_1(x+i,y+j) \times I_2(x+d+i,y+j)}{\sqrt{\sum_i \sum_j I_1^2(x+i,y+j)} \times \sqrt{\sum_i \sum_j I_2^2(x+d+i,y+j)}}$$

Two other criteria are used: C_3 and C_4. They are similar to C_1 and C_2 respectively, except for the mean gray levels value over the correlation window which is subtracted from the intensity values in the case of C_3 and C_4. The C_1 and C_3 criteria use the difference between the gray levels of the images and must be minimized The C_2 and C_4 criteria multiply the gray levels together and must be maximized. We find such normalized correlation scores C_3 and C_4 useful because they are insensitive to linear transformation of the images which may result from slightly different settings of the cameras. C_2 has performances similar to C_3 and C_4 except when the difference in the distribution of gray levels between the images is important. C_1 produces clearly worse results than the other criteria except on images with low gray level values.

5.3 Validating matches

As shown by Nishihara [17], the probability of a mismatch goes down as the size of the correlation window and the amount of texture increase. However, using large windows leads to a loss of accuracy and the possible loss of important image features. For smaller windows, the simplest definition of validity would call for a threshold on the correlation score; unfortunately such a threshold would be rather arbitrary and, in practice, hard to choose. Another approach is to build a correlation surface by computing disparity scores for a point in the neighborhood of a prospective match and checking that the surface is peaked enough [1]. It is more robust but also involves a set of relatively arbitrary thresholds.

Here we propose a definition of a valid disparity measure in which the two images play a symmetric role and that allows us to greatly reduce the probability of error even when using very small windows. We perform the correlation twice by reversing the roles of the two images and consider as valid only those matches for which the reverse correlation has fallen on the initial point in the left image.

This validity test is likely to fail in the presence of an occlusion. Let us assume that a portion of a scene is visible in the left image I_1 but not in the right image I_2. The pixels in I_1 corresponding to the occluded area in I_2 will be matched, more or less at random, to points of I_2 that correspond to different points of I_1 and are likely to be matched with them. The matches for the occluded points will therefore be declared invalid and rejected.

In fact, the density of such consistent matches in a given area of the image appears to be an excellent indicator of the quality of the stereo matching. An occasional "false positive" (a pixel for which the same erroneous disparity is measured when matching both from left to right and right to left) may occur. But, except in the presence of repetitive patterns, we have never encountered a situation that gave rise to a large clump of such errors.

When the correlation between the two images of a stereo pair is degraded our algorithm tends, instead of making mistakes, to yield sparse maps. In other words, a relatively dense disparity map is a *guarantee* that the matches are correct, at least up to the precision allowed by the resolution being used. If we reject not only invalid matches but also isolated valid matches (using a simple method based on successive erosions and dilatations) we can increase even more the ratio correct/incorrect matches without losing a large number of the correct answers.

5.4 More information about matches

It could be very interesting to really know if a validated match is reliable or not. If the match is situated of a dense area in the disparity map, the probability of a correct correlation is very high, except on repetitive patterns. But the form of the correlation curve (criterion value for all integer disparity values) shows us as well if the probability of the match to be an error is high or not. Indeed errors occur when a peak slightly higher than the right one is incorrectly chosen. So if we notice in the correlation curve several peaks with approximately the same height, the risk of choosing the wrong peak increases, especially if the images are noisy. We have therefore defined the "confidences" to be proportional to the difference of height between the two most important peaks (which must be sufficiently distant when using small windows for which the correlation curve has a lot of small noisy peaks). On repetitive patterns the correlation curve has a regular wave form and the confidences will take very low values.

We can extract another kind of information from the correlation curve. Indeed the shape of the optimal peak shows us if the matched points are situated in bland areas or not. The sharper the peak is, the more precise the localization of the matched point is. So a good way to quantify the accuracy of the sub-pixel disparity computed by the parabolic approximation is to measure the spread of the optimal peak. We assume that the peak can be locally represented as a Gaussian and take the sub-pixel precision to be proportional to the standard deviation of that Gaussian.

5.5 Hierarchical algorithm

To increase the density of our potentially sparse disparity map, we use windows of a fixed size to perform the matching at several levels of resolution (computed by subsampling Gaussian smoothed images), which is almost equivalent to matching at one level of resolution with windows of different sizes as suggested by Kanade and others [11] but computationally more efficient. More precisely, as shown by Burt and others [4], it amounts to performing the correlation using several frequency bands of the image signal.

We then merge the disparity maps by selecting, for every pixel, the highest level of resolution for which a valid disparity has been found. The reliability of our validity test allows us to deal very simply with several resolutions without having to introduce, as in [11] for example, a correction factor accounting for the fact that correlation scores for large windows tend to be inferior to those for small windows.

The computation proceeds independently at all levels of resolution and this is a departure from traditional hierarchical implementations that make use of the results generated at low resolution to guide the search at higher resolutions. While this is a good method to reduce computation time, it assumes that the results generated at low resolution are more reliable, if less precise, than those generated at high resolution; this is a questionable assumption especially in the presence of occlusions. For example in the case of tree images, it could lead to a computed distance for the area between some trunks that would be approximately the same as that of the trunks themselves, which would be wrong. Furthermore, in the absence of repetitive patterns, the output of our algorithm is not appreciably degraded by using the large disparity ranges that our approach requires.

5.6 Implementation considerations

We consider a binocular correlation on aligned horizontal epipolar lines, so that we have no vertical (y) disparities. The computation of the correlation criterion is performed in first calculating and storing the two square root terms of the denominator once and for all. Then the cross-term of the numerator is computed for each disparity value and divided by the previous square root terms to obtain the criterion value.

If we use a non normalized correlation criterion like C_2, we notice the computation of the numerator term can be simplified horizontally and vertically as below:

$$N(x,y,d) = \sum_i \sum_j I_1(x+i, y+j) \times I_2(x+d+i, y+j)$$

Let be (ni, nj) the size of the correlation window. At constant disparity, the numerator value at the point of $x+1$ abscissa can be deducted from the one at x by subtracting the contribution of the leftest column in the correlation window and adding those of the new next column, that is:

$$\begin{aligned} N(x+1,y,d) = N(x,y,d) - \textstyle\sum_j I_1(x, y+j) \times I_2(x+d, y+j) + \\ \textstyle\sum_j I_1(x+ni, y+j) \times I_2(x+d+ni, y+j) \end{aligned}$$

We have a similar relationship in the vertical direction between the numerator values at the point of y and $y+1$ ordinates:

$$N(x,y+1,d) = N(x,y,d) - \sum_i I_1(x+i,y) \times I_2(x+d+i,y) + \sum_i I_1(x+i,y+nj) \times I_2(x+d,y+nj)$$

The use of these relationships allows us to avoid any redundancies in the criterion computation and makes the processing time independent of the window size. We can also take large windows to produce more reliable results without being penalized in CPU time. There's no problem of roundness because the values I_1 and I_2 are integral.

The normalized C_3 and C_4 criteria cannot be computed in their exact form in the same way because the mean value $\overline{I}$ subtracted is estimated on the same area (the correlation window) whatever the point in the window. If we subtract first the mean value (computed on a rectangular neighbourhood of the same size as the correlation window and centered on points) to each image point and then perform the correlation using the non normalized criteria C_1 or C_2, it amounts to using the C_5 or C_6 criteria defined as below:

$$C_5(x,y,d) = \left[\sum_i\sum_j [(I_1(x+i,y+j) - \overline{I_1(x+i,y+j)}) - (I_2(x+d+i,y+j) - \overline{I_2(x+d+i,y+j)})]^2\right] / \left[\sqrt{\sum_i\sum_j [I_1(x+i,y+j) - \overline{I_1(x+i,y+j)}]^2} \times \sqrt{\sum_i\sum_j [I_2(x+d+i,y+j) - \overline{I_2(x+d+i,y+j)}]^2}\right]$$

$$C_6(x,y,d) = \left[\sum_i\sum_j [I_1(x+i,y+j) - \overline{I_1(x+i,y+j)}] \times [I_2(x+d+i,y+j) - \overline{I_2(x+d+i,y+j)}]\right] / \left[\sqrt{\sum_i\sum_j [I_1(x+i,y+j) - \overline{I_1(x+i,y+j)}]^2} \times \sqrt{\sum_i\sum_j [I_2(x+d+i,y+j) - \overline{I_2(x+d+i,y+j)}]^2}\right]$$

C_5 and C_6 yield almost the same results as C_3 and C_4 respectively; there is sometimes a little gap between scores curves but most of the time the criteria vary in the same way. The use of C_5 or C_6 allows us to have a robust normalized cross-correlation with a minimum computation time.

Acknowledgments We are especially thankful to **P. Fua**, **B.C. Bolles** and **X. Lebègue** for powerful ideas at the origin of parts of its work. Thanks to **Jean-Luc Szpyrka** for its precious help during this work. The mobile robot was a robuter builded by the French company RobotSoft. This work has been partially realized under the contract DRET 89/515.