Real time implementation of trinocular stereo vision

Grégory Randall, Serge Foret, Nicholas Ayache

UNITÉ DE RECHERCHE
INRIA-ROCQUENCOURT

Rapports de Recherche

N° 1284

Programme 6
Robotique, Image et Vision

REAL TIME IMPLEMENTATION OF TRINOCULAR STEREO VISION

Grégory RANDALL
Serge FORET
Nicholas AYACHE

Août 1990

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.:(1) 39 63 55 11

RR_1284

Real Time Implementation of Trinocular Stereo Vision [1]

Gregory Randall Serge Foret Nicholas Ayache

INRIA **NOESIS**

BP 105, *5 bis, rue du Petit Robinson,*

78153 Le Chesnay Cédex *78350 Jouy-en-Josas*

Abstract

We present the last steps achieved to implement a real time trinocular stereovision algorithm. The main improvements to the stereovision algorithm are the reduction of processing time and the diminution of the required memory. The technique has been successfully applied to several indoor and industrial scenes. Experimental results are presented and discussed.

Key-words: Stereovision, Trinocular, Edge Segments, Computer Vision, Mahanalobis Distance.

Implémentation Temps Réel de la Stéréovision Trinoculaire

Résumé

Nous présentons les dernières modifications apportées à l'algorithme de Stéréovision Trinoculaire pour son implémentation temps réel dans un système multiprocesseur.

Les améliorations principales concernent la réduction du temps de calcul et de la mémoire requise.

La technique a été appliquée avec succès à des scènes d'intérieur et industrielles. Les résultats expérimentaux sont présentées.

Mots-clefs :Stéréovision, Trinoculaire, Segments de Contour, Vision par Ordinateur, Distance de Mahanalobis.

---

[1] this work was partially supported by ESPRIT project *P*940.

1 Introduction

This paper explains the last updates made on the trinocular passive stereo vision. This research has been done for the ESPRIT Project P940 whose goals are, among others, to integrate this algorithm on a DSP 56000 board in a real time environment. The main idea of these improvements is to reduce the processing time and memory size. However, the basic philosophy keeps being the same as in the past publications:[1, 2, 3, 5]. The main modifications brougth to the algorithm are the following ones:

- better use of epipolar constraints,
- introduction of a new prediction checking test in 3rd image avoiding the neighborhood validation which was memory and time consuming.

The paper is organized as follows: we first recall background informations, developped in previous publications that are important to understand this paper. This includes geometry of trinocular stereo vision, epipolar rectification and image representation. Then we detail the matching algorithm and the validation procedure with their new developments. Finally experimental results are presented and discussed.

2 Background

2.1 Geometry of Trinocular Stereo Vision

Figure 1 illustrates the geometric constraints of trinocular stereo vision. Camera i (i = 1, 2 or 3) is represented by its optical center C_i and its image plane $\mathcal{P}_i$. Given a scene point P, its image I_i by camera i is given by the intersection of the line PC_i with the plane $\mathcal{P}_i$. This is the classical pinhole model. Points I_1, I_2 et I_3 form a triplet of *homologous* image points.

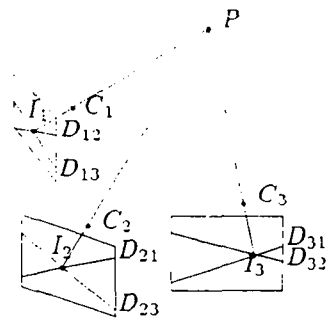

Figure 1: Geometric constraints of trinocular stereo vision

Given a pair (i, j) of cameras and a physical point P, the *epipolar plane* $\mathcal{Q}_{ij}$ is defined by the triplet of points (C_i, P, C_j). The intersection of this epipolar plane with camera plane $\mathcal{P}_i$ is the *epipolar line* D_{ij}, while its intersection with camera plane $\mathcal{P}_j$ is the epipolar line D_{ji}. D_{ij} and D_{ji} are called *conjugated epipolar lines*. Any point I_i on D_{ij} (resp. I_j on D_{ji}) has its homologous image point I_j

on D_{ji} (resp. I_i on D_{ij}). In practice the observed space is bounded in depth by Z_{min} and Z_{max} parameters. Therefore, using two cameras, the search for homologous image points is a search along conjugated epipolar intervals.

As shown on figure 1, a scene point P produces three pairs of homologous epipolar lines. When the image points (I_i, I_j, I_k) form a triplet of *homologous* image points, then I_i is necessarily located at the intersection of the epipolar lines D_{ij} and D_{ik} respectively defined by I_j and I_k. Therefore the search for homologous image points between two images can now be reduced to a simple verification at a precise location in the third image. For instance checking that (I_1, I_2) form a pair of homologous image points consists in verifying the presence of I_3 at the intersection of D_{31} and D_{32}.

2.1.1 Image representation

The matching algorithm does not perform directly on the image, but on a symbolic representation of it. For instance, we use edge segments coming from a polygonal approximation of connected chains of edge points (cf [6, 7, 9, 10, 3]).

2.1.2 Buckets

The stereo-matching algorithm access segments lying in a given region of the image. We therefore need to structure the image to optimize this operation. A very simple and efficient way to proceed is to compute buckets, i.e. superimpose a virtual grid composed of square windows on the image and compute, for each window, the list of segments intersecting it. Accessing a segment in a given region of the image is then reduced to accessing the segments of the buckets covering this area of the image. This structure is computed in linear time with respect to the number of segments. Typically, we used 32x32 buckets for the matching phase.

2.2 Rectification of images

2.2.1 Principle

For three cameras, it is possible to rectify the images to get horizontal epipolar lines between images 1 and 2, and vertical epipolar lines between images 1 and 3. In this case, the computations of epipolar segments are greatly simplified . If, in addition, the image coordinate frames are judiciously defined it is possible that the epipolar line attached to a point (u'_1, v'_1) in image 1 be the line $v'_2 = v'_1$ in image 2 and the line $u'_3 = u'_1$ in image 3. Moreover, it is possible to obtain a very simple relationship between images 2 and 3 of the form $u'_2 = v'_3$. We are then in the situation depicted by figure 2.

It appears [5, 1, 10] that rectification can be performed by linear transformations of the image coordinates in projective space by:

$$I'^{\sim}_i = R_i \, I^{\sim}_i$$

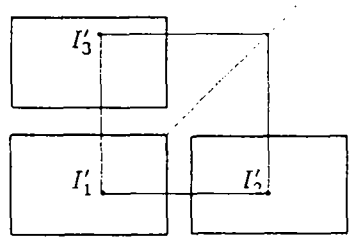

Figure 2: After the rectification of three images : the coordinates of the homologous points I'_1, I'_2 and I'_3 satisfy $v'_2 = v'_1$, $u'_3 = u'_1$ and $v'_3 = u'_2$

where the three 3x3 rectification matrices called R_1, R_2 and R_3 are defined by

$$R_i = \begin{pmatrix} (C_{i-1} \wedge C_i)^t \\ (C_i \wedge C_{i+1})^t \\ (C_1 \wedge C_2 + C_2 \wedge C_3 + C_3 \wedge C_1)^t \end{pmatrix} N_i$$

whith the conventions $i + 1 = 1$ if $i = 3$ and $i - 1 = 3$ if $i = 1$.

After the rectification of the images we have, as desired, the nice relationships:

$$\begin{aligned} v'_2 &= v'_1 \\ u'_3 &= u'_1 \\ v'_3 &= u'_2 \end{aligned} \tag{1}$$

which was illustrated by figure 2.

Since the rectification process is a linear transformation in projective space, it preserves straight lines; therefore it is sufficient to apply it to the endpoints of the linear segments of a polygonal approximation to get the endpoints of the segments of the rectified polygonal approximation. This is very useful for our stereo vision algorithms [1, 3, 4] which actually deals with linear segments.

3 Trinocular Stereo Vision

We present now the trinocular stereo vision algorithm which is divided into 3 parts:

1. Offline processing. In order to speed up the stereo matching process, following image independent steps are performed once at the begining of the process:
   - Computation of rectification matrices R_1, R_2 and R_3.
   - Computation of disparity interval related to Z_{min} and Z_{max} for each bucket.
2. Stereo matching.
3. 3D reconstruction.

Epipolar rectification and 3D reconstruction have already been discussed in previous publications (cf [5]). The main improvements have been done in the stereo matching stage, detailed in the next sections.

3.1 Improvements to the Stereo Matching Algorithm

The basic philosophy of the stereo matching algorithm is the "prediction-verification method" developed in [3]. Modifications have been carried out to reduce processing time and memory size.

The solution choosed was to improve the hypothesis generation stage by enforcing geometrical constraints and good combination of rectified and non rectified space properties in order to

- Improve the *goodmatch/wrongmatch* ratio of triplets generated by the matching algorithm.
- Avoid the neighborhood validation previously developped, which was memory and time consuming.

The first step of the algorithm generates hypothetic matches between image i and image j, this hypothesis yields to predict a 2D segment S_k^* in image k.

The problem is to compare the predicted 2D segment S_k^* with a potential match segment S_k and validate it by the following method:

1. Perform first some simple loose tests to select quickly a subset of potential matches:
   - localization (buckets).
   - orientation comparison.
   - gradient comparison.
2. Perform conservative tests using
   - colinearity.
   - overlapping.

3.1.1 Colinearity

Colinearity can be tested using the parameters (a^*,p^*) and (a,p) of the lines supporting S_k^* and S_k (see appendix). Morever, it is possible to account for the uncertainty attached to these parameters by introducing the covariance matrix $\Lambda = cov(a,p)$ within a Mahanalobis distance.

$$d^2(S_k^*, S_k) = (\Delta a, \Delta p)\, \Lambda^{-1} \begin{pmatrix} \Delta a \\ \Delta p \end{pmatrix}$$

Matrix Λ is computed from the knowledge of the uncertainty attached to the position of the endpoints of segment S_3. Typically, we assumed independent covariance of 1 pixel ^{2} for each endpoint coordinate.

The advantage of such an approach, is that a single threshold, choosen from a χ^2 table with two degrees of freedom allows for the homogeneous testing of very disparate segments (in terms of length, orientation and position). This method is similar to the one presented by Skordas et al [11].

3.1.2 Overlapping

Figure 3 illustrates 4 different data segments which can match a given predicted segment S_k^*. We assume that all of them have passed the orientation and gradient similarity tests. The colinearity test will reject segments S_a and S_b, but its posible to find segments, as S_c, which are colinear with S_k^* elsewhere.

Overlapping ratio (cf Figure 4) is calculated as follows: first, we compute $\overline{S_k}$, the projection of S_k onto the line supporting predicted segment S_k^*. Second, we compute the length of the intersection of this projection with S_k^*. Finally, the overlapping ratio is defined by the ratio between this intersection length and the length of S_k^*.

$$Overlapping(S_k^*, S_k) = \frac{||(\overline{S_k} \cap S_k^*)||}{||(S_k^*)||}$$

If the overlapping ratio is greater than a reasonable threshold, for instance 10%, segment S_k is considered as a potential match and added to the list of hypothesis.

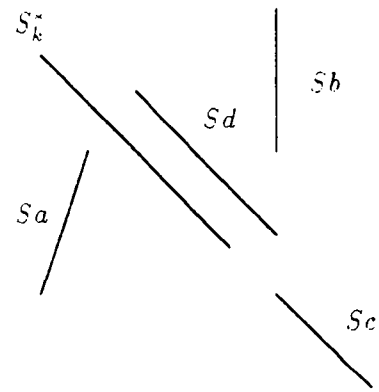

Figure 3: Discrimination of matched segments by colinearity and overlapping

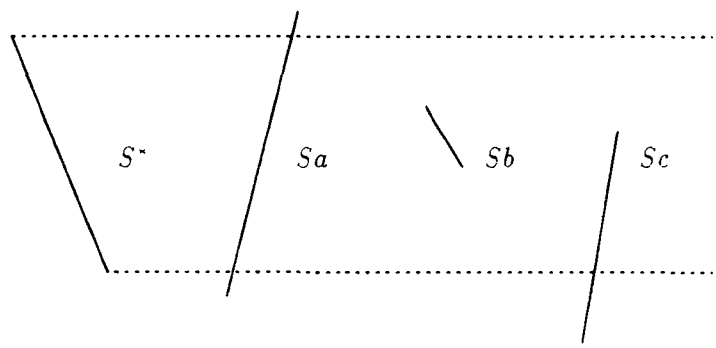

Figure 4: Overlapping ratios: $O(S^*, S_a) = 100\%$; $O(S^*, S_b) = 50\%$; $O(S^*, S_c) = 70\%$.

3.2 Stereo Matching Algorithm

The stereo matching algorithm takes as input three sets of linear segments $\{S_1\}, \{S_2\}, \{S_3\}$ coming from images 1, 2 and 3 respectively, and generates a set of triplets of matched segments $\{(S_1, S_2, S_3)\}$.

The algorithm takes advantage of the rectified space properties (cf section 2) to speed up the following steps:

- Epipolar computations. *Epipolar lines between image j and image i are horizontal, while between image k and image i they are vertical.*
- Computation of the homologous interval between segments of different images.
- Prediction of the position and orientation of a segment in the third image to validate an hypothetic match.

As geometric caracteristics of the segments are not preserved in rectified space, the geometric constraints (orientation,length,colinearity) are computed in original images.

3.3 Main Procedure

We give in figure 6 the description of the main matching procedure called STEREO-3. This procedure takes a triplet of images $(1, 2, 3)$ as input, and builds a list of matches $\{(S_1, S_2, S_3)\}$. Some on-line preprocessing is required:

- Computation of line parameters ($a.p$ and covariance matrix Λ) in images 2 and 3, used in Mahanalobis distance computation.
- Rectification of the three images.
- Structuration of rectified images 2 and 3 into buckets.

The main procedure selects every segment S_1 of the first rectified image and computes its mid-point I_1.

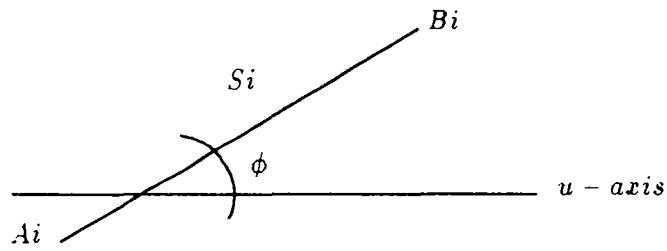

Figure 5: Parameters of one segment.

During the matching procedure, either image 2 or 3 is used first to predict hypothesis, and the other for checking . The image used to do the initial hypothesis must be such that it is the one in which the orientation of the epipolar lines is farthest from the direction of S_1.

According to the orientation ϕ_1 (the angle between the segment S_1 and the u-axis, i.e. the direction of epipolar lines, see figure 5), the procedure computes the epipolar interval D_{21} and the epipolar line D_{31} of the midpoint I_1 of S_1 in the second and third rectified images (resp. D_{31} and D_{21} in the third and second images). These intervals correspond to the possible positions of the points homologous to I_1. As images are rectified, the epipolar line is directly determined by the coordinates of I_1 and the endpoints of the epipolar interval are precalculated for each bucket related to Z_{min} and Z_{max}. For example; epipolar interval D_{21} is of the form $u_2 = u_1$, $v_2 \in [v_{min}, v_{max}]$ and epipolar line D_{31} is $v = v_1$.

The matching procedure MATCH-3, detailed in next section, returns a list of matched segments $\{(S_1, S_2, S_3)\}$ associated to S_1.

For each hypotetic triplet the homologous interval on its 3 segments is computed. This is important to detect the triplets which seems to violate the unicity constraint but in fact have one or two segments which are broken in one of the images (due to the polygonal approximation, for example).

The last part of the algorithm is the validation, which eliminates hypothetic triplets violating the unicity constraint. It is described in figure 9, and detailed in section 3.5.

3.4 Matching procedure

The algorithm of procedure MATCH-3 is given in figure 8. This procedure takes as input two rectified images j and k, a segment S_i of rectified image i, epipolar interval D_{ji} and epipolar line D_{ki}, corresponding to the midpoint I_i of S_i. It returns a list of matched triplets $\{(S_i, S_j, S_k)\}$ associated to the input segment S_i.

The first part of the algorithm is the search, in image j, for potential matches S_j of S_i. Segments S_j must

- intersect the epipolar interval D_{ji}.
- go through the similarity tests with S_i. As we have taken j such that the orientation of D_{ij} be as far as possible to the orientation of S_i, the intersection I_j between D_{ji} and S_j can in general be accurately computed. Working in rectified space allows to minimize the computation cost of the epipolar constraint test.

The second part of the algorithm checks the validity of each potential match (S_i, S_j) in image k. Using (S_i, S_j) we predict, in image k, the segment S_k^* and its associated parameters.

- the prediction of its position is made by computing the list of buckets $\{\mathcal{B}_j\}$ intersected by the predicted segment S_k^*.
- the prediction of its parameters is done in the following manner (see figure 7 for the notations):
  - Determine the rectified endpoints (A_i', B_i') , (A_j', B_j') of the homologous interval of segments S_i and S_j. This computation needs only comparisons in rectified space.

⊙⊙⊙ **Procedure** STEREO-3 (1,2,3)

For each segment in image 2 and image 3:

- Compute line parameters a,p and covariance matrix Λ.

EndFor

For each segment in image 1, image 2 and image 3:

- Compute epipolar rectification.

EndFor

For rectified image 2 and rectified image 3:

- Compute bucket structures.

EndFor

For each segment S_1 of rectified image 1.

- compute the midpoint I_1 of S_1.
- **If** S_1 is not horizontal:
  - **then**
    - determine the epipolar interval D_{21} and epipolar line D_{31}.
    - $\{(S_1, S_2, S_3)\}$ — MATCH-3$(2, 3, S_1, D_{21}, D_{31})$
  - **else**
    - determine the epipolar interval D_{31} and epipolar line D_{21}.
    - $\{(S_1, S_2, S_3)\}$ — MATCH-3$(3, 2, S_1, D_{31}, D_{21})$

EndFor

For each matched triplets (S_1, S_2, S_3)

- Determine homologous intervals on segments S_1, S_2 and S_3.

EndFor

For each of the matched triplets (S_1, S_2, S_3)

- VALIDATE-3(S_1, S_2, S_3);
  enforce the unicity constraint with a new quality criterion

EndFor

EndProcedure STEREO-3 ⊙⊙⊙

Figure 6: Trinocular Stereo Vision algorithm

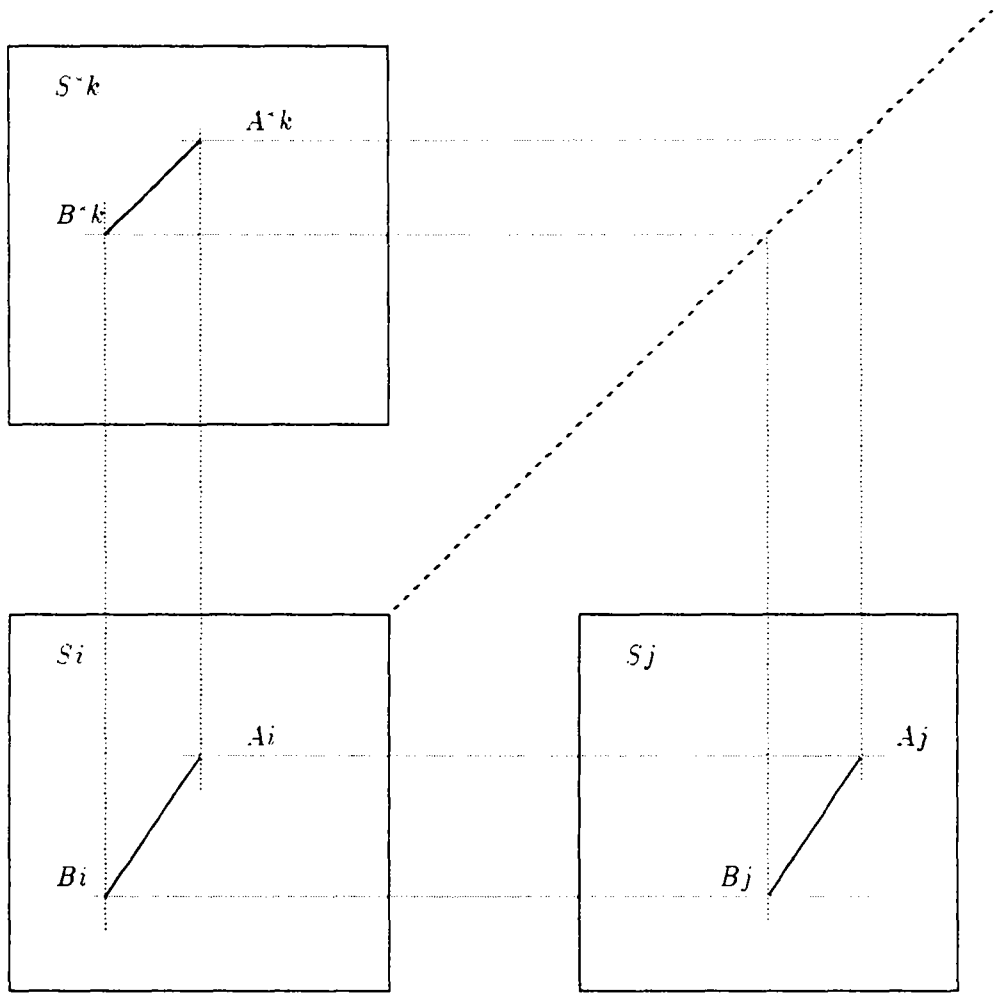

Figure 7: Prediction of segment in third image.

△ **Procedure** MATCH-3 $(j, k, S_i, D_{ji}, D_{ki})$

- Compute the list $\{\mathcal{B}_j\}$ of buckets intersected, in image j, by the epipolar interval D_{ji}.

  For each intersected bucket $\mathcal{B}_j$, determine the set of segments $\{S_j\}$ verifying the following tests :

  1. $I_j = (S_j \cap D_{ji}) \neq \emptyset$ /*Verify epipolar constraints in rectified images*/
  2. $|\phi_j - \phi_i| < \theta_{Max}$ /*Compare segments orientations in original images*/
  3. $\Delta\mu_{Min} < |Gr_j/Gr_i| < \Delta\mu_{Max}$ /*Compare average edge gradient*/
  4. $\Delta l_{Min} < |l_j/l_i| < \Delta l_{Max}$ /*Compare segments lengths in original images*/

  EndFor

  For each of the segments S_j having passed these tests,

  - Predict the segment S_k^* in rectified image k
  - Compute the following parameters of the predicted segment S_k^* in original image k:
    - Endpoints (A_k^*, B_k^*).
    - Orientation ϕ_k^*.
    - Average edge gradient Gr_k^*.
    - Supporting line parameters a_k^* and p_k^*.
  - Compute the list of buckets $\{\mathcal{B}_k\}$ intersected by S_k^*.

    For each of the segments S_k in the list of buckets $\{\mathcal{B}_k\}$, select those passing the following tests:

    1. $|\phi_k - \phi_k^*| < \theta_{Max}^*$ /*Check predicted orientation in non rectified space*/
    2. $\Delta\mu_{Min} < |Gr_k^*/Gr_k| < \Delta\mu_{Max}$ /*Check predicted gradient */
    3. $Mahanalobis(a_k, p_k, cov_{(a_k,p_k)}, a_k^*, p_k^*) < d_{Max}^2$ /*Check segments alignment*/
    4. $Overlapping(A_k^*, B_k^*, A_k, B_k) > \delta_{Min}$ /*Check segments position*/

    Endfor

  Endfor

- add $\{(S_i, S_j, S_k)\}$ to the list of hypothetic matching triplets.
- **Return**$(\{(S_i, S_j, S_k)\})$ the list of hypothetic matching triplets.

EndProcedure MATCH-3 △

Figure 8: Matching algorithm

- For each couple of endpoints (A'_i, A'_j) and (B'_i, B'_j), compute the homologous point A'_k and B'_k in rectified image k.
- The computation of the homologous point A'_k in rectified image k (see figure 7) is made as follows:
  * The first coordinate of point A'_k in rectified image k is $u_{A'_k} = u_{A'_i}$.
  * Homologous point A'_j in rectified image j is the intersection of $v_{A'_i}$ with segment S_j.
  * The second coordinate of point A'_k in rectified image k is $v_{A'_k} = u_{A'_j}$.
- Compute non rectified endpoints (A_k, B_k) in image k using inverse rectification matrix R_k^{inv}
- The predicted orientation of S_k^* is then given by the orientation of $A_k B_k$:

$$\varphi_k^* = \varphi(A_k B_k)$$

- The predicted line parameters a_k^* and p_k^* are computed from the coordinates of predicted endpoints (A_k, B_k) in original image k (see appendix).

- Knowing the predicted segment parameters; we compute the list of segments $\{S_k\}$ close to the predicted segment S_k^* using buckets. Then, for each one we check the match possibility by comparing following real and predicted segment parameters:
  - orientation in non rectified space,
  - gradient.
  - Mahanalobis distance to check colinearity,
  - overlapping to check position along a common line.

Each triplet having passed all these tests is considered as an hypothetic match and added to the list of hypothesis $\{(S_1, S_2, S_3)\}$.

3.5 Validation

The new above-described prediction scheme yields much better results than the older one. Typically, the number of false matches lies between 0 and 5 % of the total number of matches, the upper bound being attained only on very cluttered scenes. Moreover, it is now possible, due to qualitative improvements, to remove most of the errors by a simpler and cheaper final validation procedure.

It appears that most errors produce multiple matches, but not all multiple matches are errors, because a given segment can be broken differently in two or three images. Our new validation procedure detects multiple matches, and only keeps those corresponding to the correct matching of broken segments. This procedure removes almost all the errors. To do this, we enforce the unicity constraint.

First of all a list of conflicts is created by exploring the hypothesis list generated in the previous procedure MATCH-3 in order to signal which hypotetic triplets have one or two segments matched twice or more.

Figure 9 describes the validation process. We perform the validation as follows:

- we look for each segment matched twice in 2 triplets (S_1, S_2, S_3),(S'_1, S'_2, S'_3);
- if we assume that S_1 is matched twice (i.e. $S_1 = S'_1$) 2 cases can occur :
  - $S_2 = S'_2$ or $S_3 = S'_3$
  - $S_2 \neq S'_2$ and $S_3 \neq S'_3$

First case is easy to check : if $S_2 = S'_2$, the 2 triplets will be correct if S_3 and S'_3 are colinear and if there is no overlapping between them. Because of previous tests, we know that they are colinear; so we have just to check overlapping. No overlapping will mean that the segment S_1 has been cut in the third image in 2 segments S_3 and S'_3.

In the second case, the 2 triplets will be correct if they verify :

- S_2 is colinear to S'_2 and overlapping$(S_2, S'_2) = 0$
- S_3 is colinear to S'_3 and overlapping$(S_3, S'_3) = 0$

This means that this segment is cut in image 2 and 3. On the other hand, one of the 2 matches is false and we keep then the "best" one in the Mahanalobis distance sense, i.e. we keep the triplet wich was generated with a lower Mahanalobis distance in the matching phase.

This process is recursively executed in order to discard the false matches even for triplets wich have segments matched more than twice.

4 Experimental results

Table 1 gives the performance of the algorithm on several scenes. The programs are written in C and run on a SUN-3 workstation. Note that these results are obtained with fixed point arithmetics (in order to implement the algorithm in an array of DSP 56000 processors).

Memory size required has been reduced of about 30 percents. This is due to the elimination of neighbourhood validation and reduction of the number of generated triplets.

According to the scene the execution time go from half to one third of the time consumed by previous algorithm.

We have tested this algorithm on a number of industrial and office scenes, and we present now the results for a particular scene which is easy to interpret.

Figure 10 presents the 3 camera images of an indoor scene (scene 1 of table 1) and the final results of our stereo matcher and 3D reconstruction. On this above view we can see the table, the calibration grid laid on the table and some cable parts.

Figure 11 presents the input data of the stereo matcher, which are the results of the global edge detection chain: edge detection, edge linker and polygonal approximations from INRIA (cf [6, 7, 8]).

Figure 12 presents the epipolar rectification of previous images.

Figure 13 shows the reprojection of 3D segments on each image plane. These 3D segments are reconstructed from the triplets of matched segments obtained with our algorithm. Note that most of the segments are matched. However some segments are not or wrongly matched (as in the previous

Procedure VALIDATE-3 (1,2,3)

for each hypothetic triplet of segments (S_1, S_2, S_3)

for each segment of the triplet:

add to the list of conflicts.

endfor

endfor

for each conflicting triplet of segments (S_1, S_2, S_3)

for each triplet of segments (S'_1, S'_2, S'_3) such that $(S'_1, S'_2, S'_3) \neq (S_1, S_2, S_3)$ but $\exists S'_i = S_i$ (for $i = 1, 2, 3$)

if $S'_j = S_j$ (for $j = 1, 2, 3$ and $j \neq i$)

then

if $Overlapping(S'_k, S_k) = 0$

then No conflicting triplet: keep both triplets.

else Conflicting triplet: compare the Mahanalobis distance and keep the hypothesis with the lower one.

else (i.e. $S'_j \neq S_j$ and $S'_k \neq S_k$)

if

- $Overlapping(S'_j, S_j) = 0$
- $d^2(S'_j, S_j) < d^2_{max}$
- $Overlapping(S'_k, S_k) = 0$
- $d^2(S'_k, S_k) < d^2_{max}$

then No conflicting triplet: keep both triplets.

else Conflicting triplet: compare the Mahanalobis distance and keep the hypothesis with the lower one.

endfor

endfor

EndProcedure VALIDATE-3

Figure 9: Validation algorithm

	Nb. Segments			*Results*	
	image 1	*image 2*	*image 3*	*Nb. Matches*	*CPU Sun 3 Time*
Scene 1	570	554	630	317	10.2s
Scene 2	495	489	520	298	8.2s
Scene 3	329	361	349	100	4.2s
Scene 4	333	361	282	93	4.2s
Scene 5	333	369	348	107	4.4s
Scene 6	400	342	361	186	6.1s
Scene 7	404	380	365	215	6.6s

Table 1: Performance of the trinocular stereo vision algorithm on seven indoor scenes: computing time includes all processing comprising the reading of the image edge segments and the 3D reconstruction of matched triplets. Implemented on a DSP56000 multiDSP board yielded a computing time of 300 ms on scene 4.

algorithms). The main reasons are the limited precision of the preprocessing chain and the use of the gradient constraint. The strict geometric constraints eliminate some hypothesis which would be, in any case, badly reconstructed. The aim of these modifications is the implementation of the stereo matcher in a real time environment (5 Hertz), it will then be possible to eliminate the few remaining bad matches thanks to dynamic vision.

5 Conclusion

We have presented the last developments made in trinocular stereo vision. The main results are the reduction of the memory size and time consumption in order to implement the algorithm on a Multi Digital Signal Processors board(DSP 56000).

The main modifications can be summarized as follows:

- Preprocessing: all image independent computation are performed off-line.
- Hypothesis Computation : The combination of the properties of epipolar rectified and original spaces speed-up this part of the algorithm. The prediction of the segment parameters and the introduction of two new geometrical tests (i.e. colinearity and overlapping) has reduced the number of bad matches generated and allowed the elimination of the neighbourhood validation test.
- Validation: This part of the algorithm has been reduced to a unicity constraint checking. This algorithm has been developped to work in real time(5 Hertz). So, the few remaining bad matches will disapear in the next processed scene .

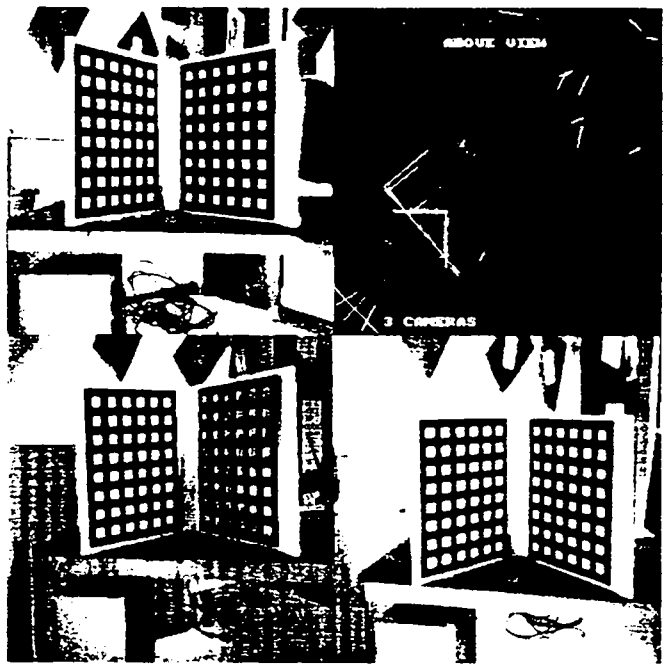

Figure 10: Triplet of images of an indoor scene and stereo vision results

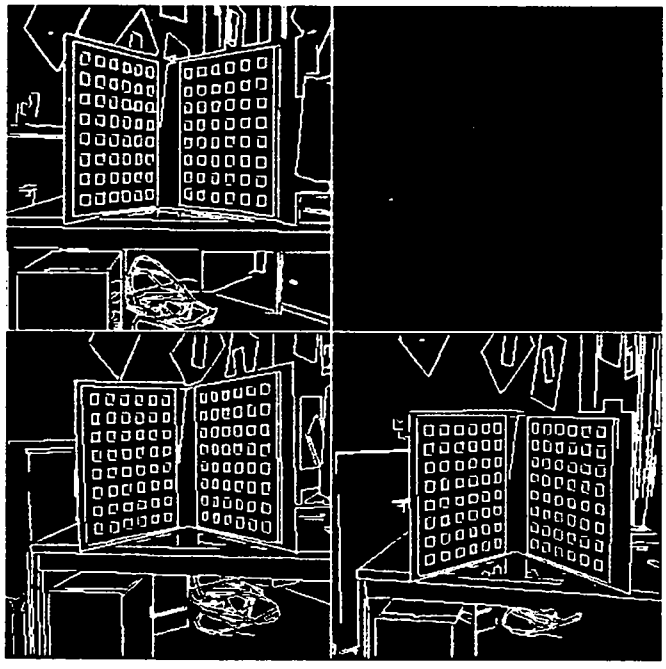

Figure 11: Polygonal approximation results.

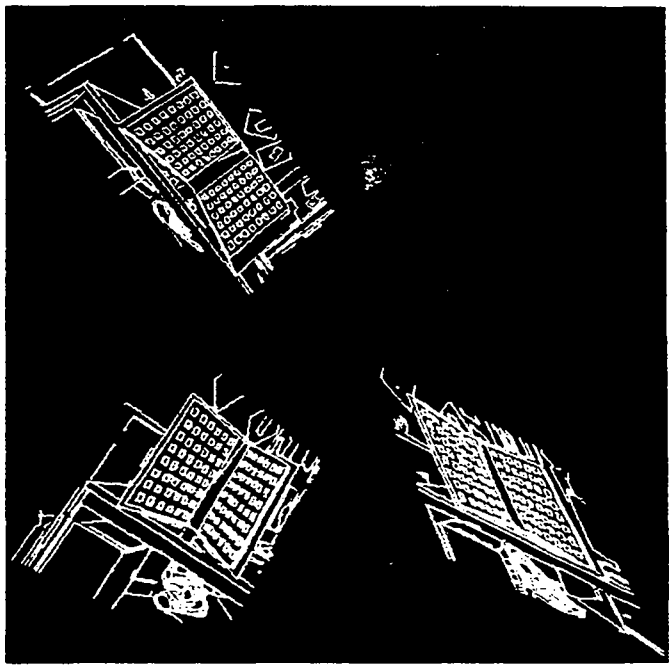

Figure 12: Epipolar rectification.

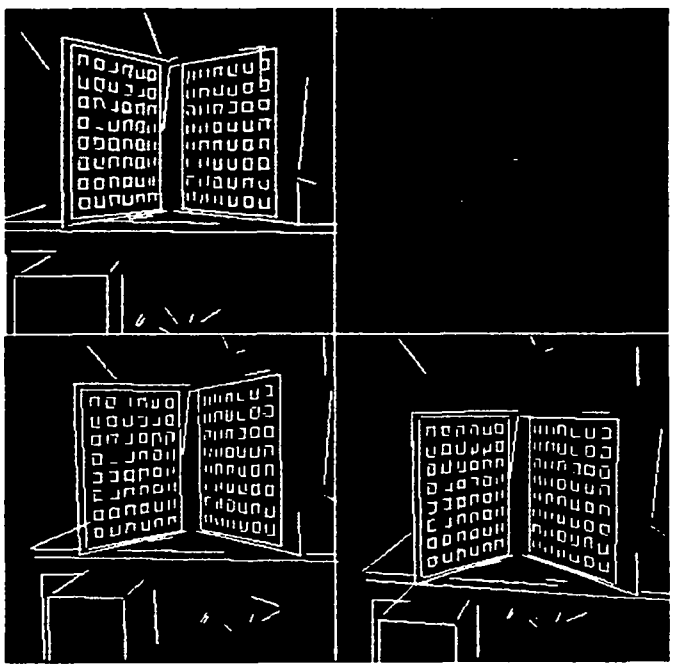

Figure 13: Reprojection of 3D segments on each image plane.

Today, the algorithm is implemented on a multiprocessor board to perform stereo matching at the rate of 3 to 5Hz, according to the image complexity. This is done within an European *ESPRIT* Project (Project P940 involving ELSAG, GEC, INRIA, MATRA, NOESIS, Univ. of Cambridge, Univ. of Genova) where preprocessing (edge extraction, edge linking, polygonal approximation) is performed by a dedicated hardware at the rate of 24 Hz.

A Mahanalobis Distance and Covariance Matrix

Let us consider a non vertical 2-D segment whose endpoints are $(x_1, y_1), (x_2, y_2)$. Given these two endpoints and their associated uncertainty ,we wish to find the parameters a and p representing the 2-D line as well as their uncertainty. These segment verify the segment line equation :

$$ax + y + p = 0$$

So, a and p coefficients can be computed from segment coordinates:

$$a = \frac{y_1 - y_2}{x_2 - x_1} \quad p = \frac{x_1 y_2 - x_2 y_1}{x_2 - x_1}$$

The covariance matrix $\Lambda = cov(a, p)$ is given by:

$$\Lambda_{a,p} = \frac{\Delta x^2 + \Delta y^2}{\Delta x^4} \begin{pmatrix} 2 & -(x_1 + x_2) \\ -(x_1 + x_2) & x_1^2 - x_2^2 \end{pmatrix}$$

The determinant of this matrix:

$$Det = \frac{(\Delta x^2 + \Delta y^2)^2}{\Delta x^6}$$

cannot be zero because we assume that the segment is non vertical (resp. non horizontal) and not a point. So, this matix is always invertible.

Let us consider $S_k = (a_k, p_k)$ one real noisy line with covariance matrices $\Lambda_{a,p}$, and $S_K^* = (a_k^*, p_k^*)$ the line supporting the predicted segment. S_k^* and S_k will be colinear if $f(S_k^*, S_k) = 0$, where f is defined as:

$$f(S_k^*, S_k) = \begin{pmatrix} a_k^* - a_k \\ p_k^* - p_k \end{pmatrix}$$

The plausibility that $f(S_k^*, S_k) = 0$ can be established by computing the generalized Mahanalobis distance $Mahanalobis(S_k^*, S_k)$ between S_k^* and S_k which is given by:

$$d^2(S_k^*, S_k) = (\Delta a, \Delta p)\, \Lambda_{a,p}^{-1} \begin{pmatrix} \Delta a \\ \Delta p \end{pmatrix}$$

As the rank of Λ is 2, the Mahanalobis distance $d2(S_k^*, S_k)$ has a χ^2 distribution whith two degrees of freedom. Looking at a χ^2 distribution table, we can choose a confidence rate by setting a Mahanalobis distance threshold :

$$d^2(S_k^*, S_k) < \chi^2_{threshold}$$

References

[1] N. Ayache. Vision Stéréoscopique et perception multisensorielle— applications à la robotique mobile. Inter-Editions, 1989.

[2] N. Ayache. Artificial Vision for mobile robots — Stereo Vision and Multisensor Perception. MIT—Press,1990.

[3] N. Ayache and F. Lustman. Fast and reliable passive trinocular stereovision. In *Proc. First International Conference on Computer Vision*, pages 422–427, IEEE, June 1987. London, U.K.

[4] N. Ayache and F. Lustman. Trinocular Stereovision for Robotics. In IEEE Trans. on PAMI, 1990. INRIA Research Rapport 1086, September 1989.

[5] N. Ayache and C. Hansen. Rectification of images for binocular and trinocular stereovision. In *9th Proc. International Conference on Pattern Recognition*, October 1988.

[6] R. Deriche. Using canny's criteria to derive an optimal edge detector recursively implemented. *The International Journal of Computer Vision*, 2, April 1987.

[7] G. Giraudon. *Chainage efficace de contour.* Rapport de Recherche 605, INRIA, Février 1987.

[8] M. Berthod. Approximation polygonale de chaines de contours. Programmes C, 1986. INRIA.

[9] G. Toscani. *Système de Calibration optique et perception du mouvement en vision artificielle.* PhD thesis, Paris-Orsay, 1987.

[10] F. Lustman. *Vision stéréoscopique et perception du mouvement en vision artificielle.* PhD thesis, Paris-Orsay, 1987.

[11] T. Skordas, P. Puget, R Zigmann and N. Ayache. Building 3-D Edge-Lines Tracked in an Image Sequence. In *Proc. Autonomous Intelligent Systems Conference* , December 1989. 2th, Amsterdam, The Netherlands.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique

ISSN 0249-6399