MVA'94 IAPR Workshop on Machine Vision Applications Dec. 13-15, 1994, Kawasaki

COOPERATION OF 3D SEGMENTS AND 3D FACETS INFORMATION FOR OBJECT RECONSTRUCTION

Didier Gemmerlé and Alain Filbois

CRIN/CNRS - INRIA Lorraine Bâtiment LORIA, Campus Scientifique, B.P. 239
54506 Vandœuvre CEDEX France *Phone: (+33) 83.59.20.67 - Fax: (+33) 83.41.30.79*
Email: gemmerle@loria.fr, filbois@loria.fr

ABSTRACT

Given a serie of trinocular images of an object, we have developed a method for building 3D Facets and 3D segments model of the object. From each triplet, a partial description of the object, called 3D View, is extracted. From the set of all extracted 3D Facets, a strategy for guiding the object reconstruction, based on a statistical method is developed. A 3D Matching Builder computes matchings between the 3D Primitives of consecutive 3D Views. Guided by the strategy, a Superstructure gatheres all the matching informations given by the 3D Matching Builder in a set of equilalence classes. For each equivalence class of Superstructure, a representative is derived. The 3D Primitives model is finally computed merging informations of 3D Facet represeentives and 3D Segment representives. This method implemented in Smalltalk80, has been applied on a series of stereoscopic real images triplets; some results are provided at the end of the paper.

1. INTRODUCTION

Reconstructing 3D objects from a set of triplets of stereoscopic images is a hard task because it involves a great number of treatments and because additive noise is very difficult to estimate. The accuracy of the reconstruction may be increased by improving the quality of the data at each step of the reconstruction [5], by combining data from different sensors or by using elaborated and informative primitives [6].

In this paper, we explain how we build a 3D model of an object using 3D Segments, i.e. couples (origin, extremity) of 3D points. In [7], 3D segments are reconstructed without taking the 2D neighbourhood of the corresponding 2D segments into account. This method is not robust when the data are noisy. In [3], a single pertinent primitive is selected in each partial description and is used as a confidence island to guide the reconstruction. We present here a method for characterizing each 3D primitive extracted from each partial description using neighbouring relations. These relations define a context which is fruitfully used to construct the model.

An object to be modelled is represented by a sequence of triplets of stereoscopic images which is supposed to visualize the principal aspects of the object. Our reconstruction system includes five main parts. The first section presents the 3D Primitive Builder which builds a partial representation of the object, called a *3D View*, in terms of the 3D segments and 3D facets extracted from a particular triplet of stereoscopic images. The second section presents the Strategy Builder which supervises the reconstruction. The next section explains how the 3D Matching Builder computes the moving between two 3D Views, using the sets of 3D primitives extracted from the two corresponding triplets. Section 5 describes the Superstructure Builder, which constructs equivalence classes of 3D primitives, and the Model Builder which selects the set of 3D primitives constituting the final model of the object. Some experimental results are presented in the last section.

2. THE 3D PRIMITIVE BUILDER

This module extracts 3D primitives from the set of triplets of stereoscopic images of the object to be modelled. There are two kinds of primitives: *segments* and *facets*.

· For each image of a triplet, an edge detection provides a set of *2D Segments* defined as pairs of 2D points. These segments are then assembled into *2D Facets* defined as ordered chains of 2D Segments surrounding an homogeneous region [2]. The 2D Facets are supposed to correspond to the projections of the facets of the 3D object on the image plane.

A stereovision algorithm [1] provides *3D Segments* using the three sets of 2D Segments extracted from each image of the triplet. They are used to build the *3D Facets*, defined as ordered chains of coplanar 3D Segments. The difficult point is to retrieve the correct ordering of the 3D segments constituting each 3D Facet. Indeed, the result of the stereovision is of poor quality because of the errors of edge detection and because of the fact that some parts which are visible in an image of the triplet are occluded in the other images. The 2D Facets of the three images of the triplet are matched so as to determine the ordering of the 3D Segments by concurrently following the ordering of the 2D segments of the 2D Facets.

If 3D Facets were only characterized by the chains of segments reepresenting their edges, the strategy builder could not efficiently control the reconstruction of the object. Quantitative information is necessary. Each 3D Facet is thus associated with a vector of attributes composed of two parts: *Intrinsic parameters*, which are primitive dependent, and *extrinsic parameters*, like the neighbouring relations, which are context dependent. For the moment, we only use intrinsic parameters:

- The surface: It is computed as the sum of the signed surfaces of all the triangles defined by the gravity centre and two consecutive vertices of the 3D Facet.
- The greatest and smallest inertia axes: They are computed from the inertia axes of all the triangles defined by the gravity centre and two consecutive vertices of the 3D Facet.
- The perimeter.
- The mean distance between the gravity centre and all the vertices of the 3D Facet.
- The maximal distance between two vertices of the 3D Facet.

3. THE STRATEGY BUILDER

It would be too time consuming to match all the possible pairs of 3D Views. Therefore, the Strategy Builder elaborates a reconstruction strategy, i.e. an order to match pairs of consecutive 3D Views, from a statistical study on the attribute vectors of all the 3D Facets of the sequence of stereoscopic triplets.

3.1. The Clusters

Each 3D Facet can be considered as a point in a 6-dimensionnal space corresponding to its attribute vector. The axis represent the greater inertia axis, the perimeter, the mean distance to the gravity centre and the maximal distance between two vertices. The 3D Facets are grouped into clusters using the algorithm of K-Means [4]. A 3D Segment is built from three matched 2D Segments all extracted from images of a stereoscopic triplet. The quality of the 3D Segment reconstruction depends on the 2D Segmentation accuracy and the relative position of each of the three camera axes of the stereoscopic triplet. In the image, an error of one pixel on the 2D Segment extraction generates an error on the length of the reconstructed 3D Segment that depends on the relative positions of the 3D Segment and the plane of the camera. If the 3D Segment is almost collinear to the axis of the camera, an error of one pixel generates an important difference on the length of the reconstructed 3D Segment. In the same way the information on the 3D Segment length is more accurate if the 3D Segment is collinear to the plane of the camera. From these observations,

the error on the computed attributes (ie, area or perimeter) is smallest when the normal to the 3D Facet is collinear to the axis of the system of trinocular camera, so we arbitrarily set the visibility angle of the 3D Facet to 90 degrees. The number of clusters that we wish to find is $n_c = \frac{3DViewnumber}{4}$. The method compute n clusters, the elements of which are neighbors with the usual Euclidean distance. From these 3D Facets clusters we deduce 3D Views clusters as follows.

Each 3D Facet cluster correspond to a 3D View cluster, defined as the set of 3D Views in which the 3D Facets of the cluster appear. For example, let a 3D Facet cluster be $\{F_1^2, F_3^2, F_3^3, F_1^4, F_2^6\}$, where F_i^j is the i-th 3D Facet in 3D View number j. The corresponding 3D View cluster is $\{V_2, V_3, V_4, V_6\}$, where V_i is 3D View number i. A 3D View cluster includes the 3D Views comprising 3D Facets which are similar according to the distance used in previous section.

3.2. The Reconstruction Strategy

The reconstruction of the object is based on the matchings of 3D Primitives which belong to two consecutive 3D Views of the sequence. Two matched primitives are supposed to represent the same physical entity. The Strategy Builder has to find the set of 3D Views representing the best candidates to start the reconstruction, and then to determine how to match the successive 3D Views in order to obtain the 3D structure of the object.

Each 3D View of a 3D View cluster is associated with a weightaccording to the "position" of the 3D View in the cluster, considering only series of consecutive 3D Views. For example, let 3D View cluster I be $\{V_2, V_3, V_4, V_5, V_6, V_8, V_9\}$ (Fig. 1). The 3D View V_3 has rank 2 in the series of five consecutive 3D Views $\{V_2, V_3, V_4, V_5, V_6\}$ in which it is included. The local weight of the View is computed using its rank i and the length l of the series in the following way: Weigth(i,l) =

$$\begin{cases} i & \text{if } 1 \leqslant i < l/2 \\ l-i+1 & \text{if } l/2 \leqslant i \leqslant l \\ 0 & \text{otherwise} \end{cases}$$

The global weight of a 3D View is defined as the sum of the local weights of the 3D Views in each cluster in which the 3D View appear. Thus, in the example, the global weight of 3D View V_5 is 3.

A similarity between the 3D Facets belonging to close 3D Views implies that these 3D Views have a high weight

3D View cluster \ 3D View number	1	2	3	4	5	6	7	8	9	10	11	12
I		1	2	3	2	1		1	1			
II	2	1										1
III					1	2	3	4	3	2	1	
total weight	2	2	2	3	3	3	3	5	4	2	1	1

Figure 1. The computation of the weight of a 3D View.

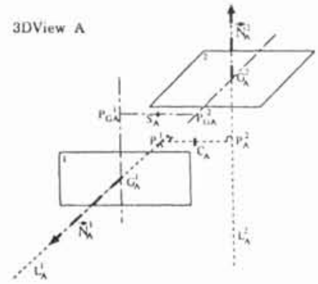


Figure 2. Computation of the 3D View Centre.

as they belong to several clusters. So, the 3D View associated with the highest weight, 3D View number 8 in the example, is used to start the reconstruction of the object. The order of the matchings between consecutive 3D Views is determined by the longest series of 3D Views in which the highest weighted 3D View appear. This series is scrutinized from left to right and then from right to left, starting with the highest weighted 3D View. In the example, the order of the matchings is $(V_8,V_9)(V_9,V_10)(V_{10},V_{11})(V_8,V_7)$ $(V_7,V_6)(V_6,V_5)$. The next matchings are determined using the longest series which have a common 3D View with the last processed series. If such a series does not exist, the series having the highest weighted 3D View is selected. This process is repeated until every 3D View appears in one matching at least.

Then, the Strategy Builder invokes the 3D Matching Builder to perform the matchings in the order which has been established.

4. THE 3D MATCHING BUILDER

Currently, the matching between the 3D primitives of two consecutive 3D Views is not implemented and we have made the matchings manually. So, at this stage, we suppose that the matching between the 3D primitives of 3D View A and 3D View B is known. The motion between the two 3D Views is computed in three steps, supposing that the object is rigid. We first determine reference 3D Points in each 3D View. These points are used to compute the rotation the rotation and then the translation between the 3D Views.

From these information, the expression of the transformation of a 3D Point in 3D View A to its corresponding point in 3D View B is: $C_B + \overrightarrow{T_{AB}} + R_{AB}\ (X_A - C_A)$ where C_A (resp. C_B) is the 3D View A (resp. B) Centre , $\overrightarrow{T_{AB}}$ is the translation between 3D View A and 3D View B, and R_{AB} is the rotation between 3D View A and 3D View B.

4.0.1. Computation of the 3D View Centre

In each 3D View, the reference 3D Point C_A (resp. C_B), called *Centre of View* (Fig. 2), is computed as follows.

Let $3DFacetSet_A$ (resp. $3DFacetSet_B$) be the set of 3D Facets in 3D View A (resp. 3D View B). Two 3D Facets F_A^1 and F_A^2 are chosen in $3DFacetSet_A$ so that:
bullet the angle between their normals is as close to $\frac{\pi}{2}$ as possible
bullet F_A^1 and F_A^2 matches respectively F_B^1 and F_B^2 in $3DFacetSet_B$
Let N_i^j ($i \in \{A,B\}$ and $j \in \{1,2\}$) be a vector orthogonal to F_i^j and G_i^j ($i \in \{A,B\}$ and $j \in \{1,2\}$) the gravity centre of F_i^j. Let L_i^j ($i \in \{A,B\}$ and $j \in \{1,2\}$) the line passing by G_i^j and having N_i^j as supporting vector. We compute the 3D Segment $[P_i^1, P_i^2]$ as L_i^j passing by P_i^j ($i \in \{A,B\}$ and $j \in \{1,2\}$) and $[P_i^1, P_i^2]$ orthogonal to L_i^1 and L_i^2 ($i \in \{A,B\}$).
C_A (resp. C_B) is defined as the middle of $[P_A^1, P_A^2]$ (resp. $[P_B^1, P_B^2]$).
From C_A and C_B we compute the vector rotation between View A and View B.

4.1. The Computation of the Rotation

The vectorial rotation R_{AB} is computed using Kim and Aggarwal's method described in [8]. The rotation is represented by the vector of the rotation axis $\vec{V}$ and an angle θ so that: $\theta = \|\vec{V}\|$ where $\vec{V}$ is given by:

$$\vec{V} = \frac{(\overrightarrow{C_B G_B^1} - \overrightarrow{C_A G_A^1}) \wedge (\overrightarrow{C_B G_B^2} - \overrightarrow{C_A G_A^2})}{\|(\overrightarrow{C_B G_B^1} - \overrightarrow{C_A G_A^1}) \wedge (\overrightarrow{C_B G_B^2} - \overrightarrow{C_A G_A^2})\|} \text{ and}$$

$$\cos(\theta) = \frac{(\vec{V} \wedge \overrightarrow{C_A G_A^1}).(\vec{V} \wedge \overrightarrow{C_B G_B^1})}{\|\vec{V} \wedge \overrightarrow{C_A G_A^1}\|.\|\vec{V} \wedge \overrightarrow{C_B G_B^1}\|}$$

Let $\vec{V} = (V_x, V_y, V_z)$. We have:

$$M(\vec{V}) = \begin{pmatrix} 0 & -n_z & n_y \\ n_z & 0 & -n_x \\ -n_y & n_x & 0 \end{pmatrix}$$

and

$$R_{AB} = I + sin(\theta)M(\vec{V}) + (1 - cos(\theta))M(\vec{V})^2$$

The nest step is the computation of the translation between 3D View A to 3D View B.

4.1.1. The Computation of the Translation

The error on the coordinates of the 3D Points C_A and C_B (cf. Fig. 2) depends on the accuracy of the computation of the normals, which we do not know how to estimate for the moment. The translation is thus computed using two new 3D Points S_A and S_B. Let P_{GA}^1 (resp. P_{GA}^2) be the projection of G_A^1 (resp. G_A^2) on Facet F_A^2 (resp. F_A^1) and P_{GB}^1 (resp. P_{GB}^2) be the projection of G_B^1 (resp. G_B^2) on Facet F_B^2 (resp. F_B^1). Let S_A (resp. S_B) be the middle of the segment $[P_{GA}^1, P_{GA}^2]$ (resp. $[P_{GB}^1, P_{GB}^2]$). So, we define the translation as follows:

$$\overrightarrow{T_{AB}} = \overrightarrow{C_B + R_{AB}(S_A - C_A), S_B}$$

We present now the Superstructure, the aim of which is to gather the information given by the 3D Matching Builder.

5. THE SUPERSTRUCTURE

The information supplied by the 3D Primitive Builder and the 3D Matching Builder (3D Facet matchings, 3D Segment matchings and rotations and translations between 3D Views) is grouped in the *Superstructure*, which is used to construct the global model of the object.

The primitives are divided into equivalence classes: Two primitives belong to the same class if they have been matched by the 3D Matching Builder. As there are two kinds of primitives, facets and segments, the Superstructure comprises two parts: The 3D Segment Superstructure contains the equivalence classes of 3D Segments whereas the 3D Facet Superstructure contains the equivalence classes of 3D Facets (Fig. 3). A 3D Segment is represented by the two 3D Points constituting its extremities. A 3D Facet is represented by the list of the segments constituting its edges, its gravity centre and the normal to the least square plane of the facet. Each equivalence class is associated with a element representative. The coordinates of a 3D Segment representative are the mean coordinates of the extremities of all the 3D Segments of the corresponding class. The normal (resp. gravity centre) of a 3D Facet representative is the mean normal (resp. gravity centre) of all the 3D Facets of the corresponding class.

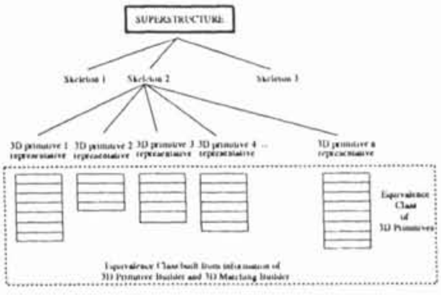


Figure 3. Structure of the 3D Facet (3D Segment) Superstructure.

It is possible that two consecutive 3D Views have not been matched, because, for example, the aspect of the object changes between the two views. The coordinates of all the 3D Primitives of a series of consecutive matched 3D Views are expressed in the frame of a unique 3D View of the series, chosen as reference. The elements representative of these primitives constitute a so called *Skeleton* of the object, i.e. the structure of a part of the object, corresponding to a given angle of view (cf. Fig. 3).

6. THE MODEL

A 3D Facet can be viewed as an ordered set of 3D Segments, but also as a set of 3D Segments verifying the same property of coplanarity (ie. the coordinates of the 3D Segments extremities verify the following equation: ax+by+cz+d=0). This second point of view makes it easier to formalize the 3D reconstruction of the object. The model of the object is constituted by the 3D Segments representative transformed so that these coordinates of 3D Segments extremities verify the equation that characterize the least means square plane computed from the set of 3D Points of the 3D Facet. We have developed a method for computing the plane of a 3D Facet from its set of coplanar 3D Facets. Here we present only the computation of the 3D Segments of the Model and we suppose that the 3D Facets of the Model are the 3D Facets representatives of the Superstructure (cf section 5). We sketch now our method for computing the 3D Segments of the Model. Let $Reps_S$ be the set of 3D Segments representative and Rep_F be the set of 3D Facets representative. Let $Class_r$ denote the equivalence class r is the element representative of which. Let $LSeg_f$ be the list of 3D Segments of the 3D Facet f and $\mathcal{P}_f(s)$ be the projection of the segment s on f. The set of 3D Segments S_{Mod} of the model $SegMod$ is computed in the following way (Fig. 4): For each 3D Segment representative S_{rep} in Rep_S, we first look for a 3D Segment (*3D Segment k*) such that:

- *3D Segment k* is in the $Class_{S_{rep}}$, and

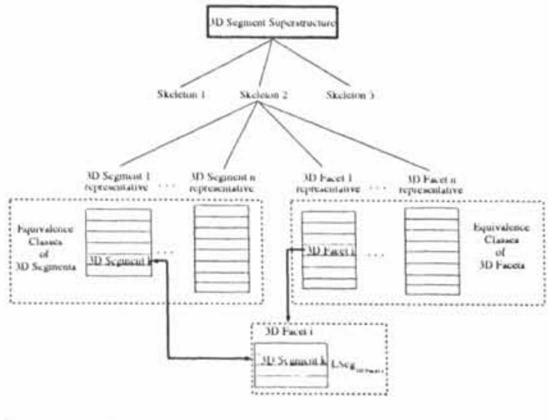


Figure 4. Computation of the 3D Segments of the model.

- there exists a 3D Facet representative F_{rep} in Rep_F and a *3D Facet i* in $Class_{F_{rep}}$ such that *3D Segment k* is in $LSeg_{3D\ Facet\ i}$

If a such *3D Segment k* exists then the 3D Segment S_{Mod} in $SegMod$ is $\mathcal{P}_{F_{rep}}(S_{rep})$; otherwise, S_{Mod} is S_{rep}.

In this way, the contextual information constituted by the 3D Facets representative and the intrinsic information constituted by the reconstructed 3D Segments are merged. Therefore, a set of 3D Segments detected as coplanar by the 3D Facet Builder and incorrectly reconstructed can be adjusted. We are currently studying more precisely how to solve the inconsistencies of the information contained in the Superstructures.

7. EXPERIMENTAL RESULTS

Our system have been experimented on series of calibrated triplets of stereoscopic images. We present here an example of results on a serie of 20 triplets. The triplets have been irregularly shot during a complete revolution around the object. The object, a cardboard box, is rather simple because the implementation of the matching modules is not yet achieved. Their results have been simulated.

The 3D Builder, implemented in C language, constructs a representation (a 3D View) of every triplet in terms of 3D Facets and 3D Segments (Fig. 5). 65 3D

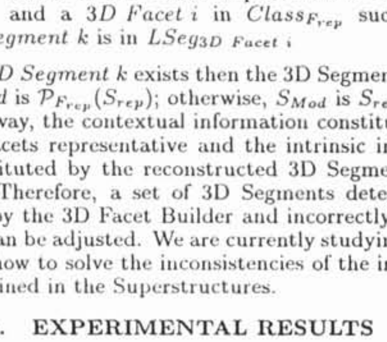
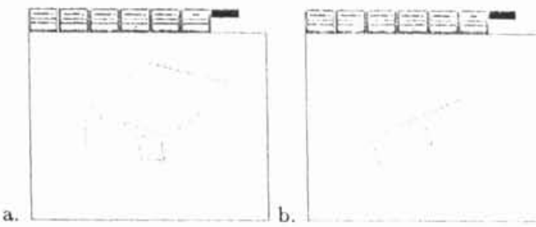


Figure 5. A sample of 3D Views

Facets have been extracted from the 20 3D Views. After computation of their attribute vectors, the facets have been divided into 4 clusters using the K-mean algorithm. Clusters 1, 2, 3, 4 and 5 include respectively 22, 7, 4, 15, and 17 3D Facets. The reconstruction performed according to the order established by the Strategy Builder has provided 4 Skeletons (Fig. 7). The Superstructure and Model builders are been implemented in Smalltalk80.



a. b.

Figures 7.a and 7.b show that the quality of the result is almost acceptable when the reconstructed 3D Segments are not too much noisy. The 3D Facet information forces the planarity of a set of 3D Segments include in same 3D Facet (cf. Fig. 7.b). When the reconstructed 3D Segments are noisy, they cannot be merged into the 3D Facets.It seems necessary to use extra contextual information, such as groups of connected segments or groups of colinear segments.

8. CONCLUSION

The originality of our approach is the use of 3D Facets as global 3D primitives but also as contextual information about the 3D Segments provided by the stereovision process. The relations of coplanarity between segments is propagated through all the successive steps of the reconstruction, until the constitution of the final model.

We must now pay a particular attention to three main points:

- Introduce new 3D primitives, such as colinear 3D Segments and coplanar 3D Facets, to be efficiently used during the construction of the final model.
- Reconstruct 3D Segments (i.e. edges of the object) by intersecting 3D Facets, when it is possible.
- To be able to solve the inconsistencies of the information of the Superstructure and to revise the division into equivalence classes.

The second point strongly depends on the third point, to which our future effort must be devoted. The redundancy of the information provided by the Superstructure might allow us to obtain much better results.

REFERENCES

[1] N. Ayache. *Vision stéréoscopique et perception multisensorielle. Applications à la robotique mobile.* InterEditions, 1989.

[2] H. Chabbi and G. Masini. A Combined Use of Regions and Segments to Construct Facets. In *International Conference on Image Analysis and Processing, Como (Italy)*, pages 334–338, september 1991.

[3] K. Ikeuchi and K.S. Hong. Determining Linear Shape Change: Toward Automatic Generation of Object Recognition Programs. *CVGIP: Image Understanding*, 53(2):154–170, 1991.

[4] Jain, A.K., and R. Dubes. *Algorithms for Clustering Data.* Prentice Hall, Englewood Cliffs, N.J., 1988.

[5] R. Mohr, B. Boufama, and P. Brand. Accurate relative positioning from multiple images, 1993. submitted to Artificial Intelligence Journal.

[6] M. Thonnat. Semantic Interpretation of 3D Stereo Data: Finding the main structures. In *Proceedings of 8th International Conference on Pattern Recognition, Paris (France)*, pages 1051–1054, 1986.

[7] A. Winzen and H. Niemann. Automatic Model-Generation for image Analysis. *Graphics and Robotics*, 1993.

[8] Z. Zhang. *Motion Analysis From A Sequence Of Stereo Frames And Its Application.* Thèse de doctorat, University of Paris XI, october 1990.