Multi-Criteria Data Association in 3D-6DoF Hierarchical SLAM with 3D Segments

Daniele Marzorati
Università di Milano-Bicocca
Milano, Italy
marzorati@disco.unimib.it

Matteo Matteucci
Politecnico di Milano
Milano, Italy
matteucci@elet.polimi.it

Domenico G. Sorrenti
Università di Milano-Bicocca
Milano, Italy
sorrenti@disco.unimib.it

Abstract

This work deals with the Simultaneous Localization and Mapping (SLAM) problem when the data sensed are 3D segments. Vision systems generating 3D data (points or segments) are quite reliable and in use since many years in the robotic and computer vision community, but SLAM with 3D vision is not yet common, although necessary for service robot activities. This paper deals with two of the issues that prevent the diffusion of this approach: data association and the related accuracy of estimates. Our claims are: 1) robot is immersed in 3D environment and using redundant sensing systems is expansive; 2) the use of different data-association criteria improves the search and turns into less data association errors; 3) reliable 3D world mapping can be obtained in large indoor environments using our proposed improvements. Experimental activity on real data confirms our claims.

1 Introduction

Some robot activity requires a full 3D knowledge of the observed environment features; a few examples are: tables legs or steps which constrain motion; door handles and fire extinguishers help in localization; cleaning has to be performed under tables and chairs; fire extinguishers have to be avoided; books to be moved are on top of tables, etc. (see Figures 1 and 2). Most of these items are not perceivable with a 2D laser range finder (LRF). It is therefore relevant to map the real 3D robot workspace. This has been not so common up to now in SLAM systems; most of the works (e.g., [14]) dealing with 3D data bases on 3D LRFs. These devices provide just geometric data, which makes difficult other robot tasks; an example is the semantic classification of places [12], which are required for a real indoor service robot. Even though we are here proposing to use just the geometry provided by 3D vision systems (because we are working with the geometric task of map building), we think that the full richness of vision systems output is necessary for other tasks. Moreover, as pointed out by [1], an indoor robot with realistic sale expectations cannot base on a costly sensing suite; in his presentation, C. Angle reported a, maybe provoking, estimate of 10US$ cost for an all-inclusive complex robot sensor. 3D vision-based sensing includes the capabilities obtainable from costly (3D or 2D) LRFs. We therefore think we need to deal with robotics tasks like SLAM with 3D data from vision.

Figure 1: Robot explores/cleanes under the table (it can do it only with 3D reconstruction).

In Section 2 we shortly review the used 3D sensing system. In Section 3 we give some details about the 3D six Degree of Freedom (6DoF) Hierarchical SLAM, while in Section 4 we present a state-of-the-art of data-association algorithms (Joint Compatibility Branch and Bound [10]). Section 5 introduces fundamental improvements in data association, number of pose DoF, and filtering which could be of interest also for 2D-3Dof SLAM (included the Iterative Closest Point [4] method). Experiments are then presented to instantiate the claims.

Figure 2: Robot is in collision course with the opened window (it can avoid it only with 3D reconstruction).

2 3D Vision Sensing System

Several cheap and fast 3D vision system are available in the commercial and academic computer vision community like the stereo vision systems, trinocular vision systems [2], and correlation-based SRI's Stereo Engine [8]. Since we consider 3D segments as "natural landmarks" in indoor environments, in our paper we focus on the trinocular approach [2]; it deals with 3D segments from the very first processing step. The process starts with the acquisition of the images from the three cameras. The three cameras are calibrated with respect to a common reference system, located on the robot. We use a DLT technique to determine the three projection matrices. These matrices are necessary for the following steps. A features extraction process generates a set of elements (2D segments in this case) from the images. These segments are described by the coordinates of their extrema ($[x_1, y_1, x_2, y_2]$). Next, the trinocular approach bases on geometric trinocular constraint ([2]), to speed up the search for the corresponding segments. The last stage is the computation of the 3D segments parameters. The 3D support line is first calculated from triplet of corresponding segments. Then the 3D extrema are computed using the interpretation lines from each 2D segment; the 3D segment given in out is the intersection of all such 3D reprojected segments). The 3D segments are represented by the 3D coordinates of their endpoints, as well as an associated uncertainty.

In Figure 3, $\mathbf{D}$ is the 3D scene segment, $\mathbf{C_i}$ and $\mathbf{d_i}$ are respectively the projection center and the projection of $\mathbf{D}$ for image i. Cameras are calibrated altogether with their covariance matrix so that 3D segment endpoints can be given out with an associated covariance matrix, to represent the measurement uncertainty as a normal probability distribution. Such systems date a long time ago and are quite widespread in the computer vision and robotics communities. Our implementation differs from the original only in the use of the Fast Line Finder [7] in the polygonal approximation phase.

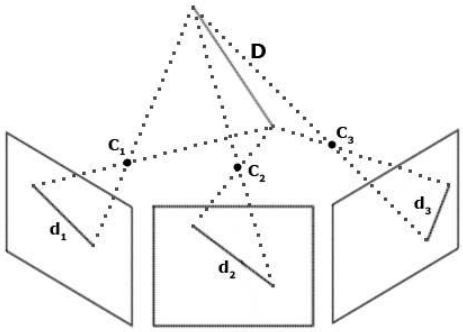

Figure 3: 3D segment-based reconstruction for a trinocular stereoscopic system.

3 3D 6DoF Hierachical SLAM

In this section we describe our SLAM system. Our proposal bases on a local and a global world model (Hierarchial SLAM [6]). The two represent, in a sense, two abstraction levels we can have in considering the space. The global level is based on the local one, i.e. it is not useful alone. The local level is represented by a set of submaps of limited cardinality, which represent of a subset of the world features. At the global level a graph-based representation of the world is introduced: the submaps are nodes while edges represent the (pose) constraints between nodes. In our implementation we use a 3D-6Dof approach where the first item referst to the dimensionality of the data used for building the world model (we have 3D data from the perception system), the second item refers to the degrees of freedom of the robot pose (full rigid-body transformation). More details can be found in [9].

Now we explain the notation used in the following sections. v_i represent the coordinates of the observed segment (two 3D extrema):

$$v_i = [x_i', y_i', z_i', x_i'', y_i'', z_i'']^T \quad (1)$$

The covariance of this segment is:

$$\mathbf{P}_{v_i} = \begin{bmatrix} \sigma^2_{x_i'} & \sigma^2_{x_i'y_i'} & \cdots & \sigma^2_{x_i'z_i''} \\ \sigma^2_{y_i'x_i'} & \sigma^2_{y_i'} & \cdots & \sigma^2_{y_i'z_i''} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma^2_{z_i''x_i'} & \sigma^2_{z_i''y_i'} & \cdots & \sigma^2_{z_i''} \end{bmatrix} \quad (2)$$

The m_j segments are described similarly.

4 Data-Association Algorithms

In most indoor applications, the robot pose is represented by the $< x, y, \vartheta >$ triplet of DoFs. With exception of some recent works [13, 14, 9], on which we elaborate further in the sequel, this accounts for the large part of indoor SLAM systems.

In the following we introduce a simple formalization of the data association issue to be used in the rest of the paper. Suppose we have measured a set of features $\mathcal{V} = \{v_i\}$ from a certain robot position. Each feature is represented by its coordinates and its uncertainty. We have also the set of features $\mathcal{M} = \{m_j\}$ observed and integrated in the map up to that moment. The purpose of a data-association algorithm is to find a proper hypothesis of correspondences between them (list of couples v_i and m_j) $H_k = \{< v_1, m_{j_1} >, < v_2, m_{j_2} >, \ldots, < v_V, m_{j_V} >\}$. To generate this hypothesis we must explore an interpretation tree where each node represents a couple of corresponding features $< v_i, m_j >$ (see Figure 4).

One of the earliest algorithms for data association in the *nearest neighbor* matching algorithm presented in [3]; it proceeds as follows:

1. take v_i from $\mathcal{V}$ and move it into the $\mathcal{M}$ reference frame
2. compute a match measure for v_i, with respect to all $m_j \in \mathcal{M}$
3. select $< v_i, m_j >$ if m_j is the best match for v_i *and* the match measure is better than a given threshold
4. erase v_i from $\mathcal{V}$ *and* m_j from $\mathcal{M}$
5. go back, and iterate with v_{i+1}.

In this method each feature in the view is associated to a map feature independently on the others. The comparison between features from the two sets are made once for all, without the possibility to find a better match, for a map feature, with another view feature (change order v_i out from $\mathcal{V}$ changes the result of the algorithm). This turns out to be a greedy algorithm to traverse the interpretation tree of Figure 4. The v_i segment is moved into the Map reference frame (*step1*) by means of an estimate of the robot pose, e.g. the one provided by odometry. Such estimate is known altogether with an estimate of its uncertainty, which is propagated to the segment in the Map reference frame. The data association criteria in use in this system is based on the Mahalanobis distance of the extrema of the 2 segments, $< v_i, m_j >$, in the map reference frame (*step2*). In this case, the point-to-point Euclidean distance $D_P(\mathbf{P_1}, \mathbf{P_2})$ is used.

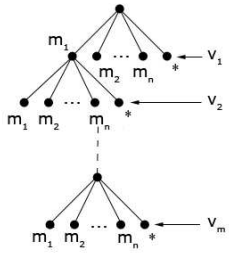

Figure 4: Interpretation tree of features $\mathcal{M}$ and $\mathcal{V}$

Being v_i' the 1st extremum of segment v_i and v_i'' the 2nd, we have to deal with the distance of the two extrema of the segments, therefore the considered distance is:

$$\mathbf{h}(v_i, m_j) = \begin{bmatrix} D_P(v_i', m_j') & D_P(v_i'', m_j'') \end{bmatrix}^T \quad (3)$$

It is now possible to compute the Mahalanobis distance between the two segments and compare it with a given threshold[1]:

$$D^2_{\mathbf{h}} = \mathbf{h}^T \mathbf{C}^{-1}_{\mathbf{h}} \mathbf{h} < \chi^2_{d,\alpha} \quad (4)$$

where the covariance $\mathbf{C_h}$ of $\mathbf{h}$ is obtained as:

$$\mathbf{C_h} = \mathbf{H}\mathbf{P}_{m_j}\mathbf{H^T} + \mathbf{G}\mathbf{P}_{v_i}\mathbf{G^T} \quad (5)$$

where:

$$\mathbf{H_{(2,6)}} = \mathbf{J}\frac{\mathbf{h}}{\mathcal{M}} = \begin{bmatrix} \frac{\partial D_P(v_i', m_j')}{\partial \mathcal{M}} & \frac{\partial D_P(v_i'', m_j'')}{\partial \mathcal{M}} \end{bmatrix}^{\mathbf{T}}$$

[1] The Mahalanobis distance measures the similarity between two sets (d-dimensional). This threshold is the probability α (confidence level) to accept false matches.

$$\mathbf{G}_{(\mathbf{2},\mathbf{6})} = \mathbf{J}\frac{\mathbf{h}}{\mathcal{V}} = \left[\begin{array}{cc} \frac{\partial D_P(v_i',m_j')}{\partial \mathcal{V}} & \frac{\partial D_P(v_i'',m_j'')}{\partial \mathcal{V}} \end{array} \right]^{\mathbf{T}}$$

A significant advancement w.r.t. nearest neighbor is the *joint compatibility branch and bound* association algorithm that allows considering, in each data-association iteration, all feature association choices computed until that time and the correlation between them (see [10] for details). Indeed, the measurements are always correlated by the uncertainly on the robot position (this information is lost when we use a classical nearest neighbor approach). In this way it is possible to reconsider the previous couples of features to limit the possibility of accepting false matching. This is obtained by traversing the interpretation tree using a branch and bound method in conjunction with a joint compatibility test. The quality of a node is represented by the number of non-null association executed to reach that node. Consider the following scoring function $\mathbf{f}_{H_k}$:

$$\mathbf{f}_{H_k} = \left(\begin{array}{c} \mathbf{f}(v_1, m_{j_1}) \\ \vdots \\ \mathbf{f}(v_V, m_{j_V}) \end{array} \right)$$

The joint compatibility of a hypothesis H_k is determined by:

$$\mathbf{C}_{H_k} = \mathbf{H}_{H_k}\mathbf{P}\mathbf{H}_{H_k}^{\mathbf{T}} + \mathbf{G}_{H_k}\mathbf{S}\mathbf{G}_{H_k}^{\mathbf{T}} \quad (6)$$

$$D_{H_k}^2 = \mathbf{f}_{H_k}^{\mathbf{T}}\mathbf{C}_{H_k}^{-\mathbf{1}}\mathbf{f}_{H_k} < \chi_{d,\alpha}^2 \quad (7)$$

where $\mathbf{P}$ is the covariance of the state (robot and features pose), $\mathbf{S}$ is the covariance of the observations and:

$$\mathbf{H}_{H_k} = \frac{\partial \mathbf{f}_{H_k}}{\partial \mathcal{M}} \quad (8)$$

$$\mathbf{G}_{H_k} = \frac{\partial \mathbf{f}_{H_k}}{\partial \mathcal{V}} \quad (9)$$

Please refer to [10] for details about the branch and bound technique.

5 Advances in 3D Matching

Our claim in this paper is that one of the key points in the data association phase is the definition of a proper criterium to match features in the view with features in the map. Usually the point-to-point distance is considered as an appropriate criterium for single segment matching and much of the effort is devoted in finding a good association strategy for dealing with the exponential complexity of finding the best match

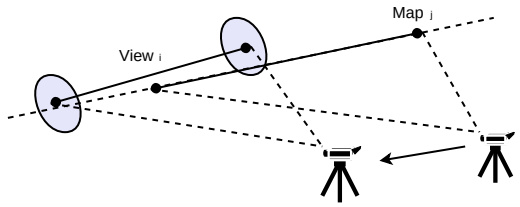

Figure 5: The "moving window" problem.

for the whole view. We will show in the following that defining a better criterium for 3D segment matching will result in a better data association almost independent from the algorithm for interpretation tree traversal (i.e., data association). The approach we propose is based on a multi-criteria evaluation for associating segments in the View with segments belonging to the Map. The reason for this change from the point-to-point criterium is mainly due to the problem of moving-field-of-view in the sensing system, which turns in a moving window on the world feature, see Figure 5. More precisely the segments extrema are introduced by the reduced field of view of the cameras and are not always related to real extrema in the world; having a sensing system that moves introduce new extrema, and thus new possible segments, at each step and this can easily become a problem for the classical point-to-point distance. Obviously our proposal is of interest also for 2D-3DoF SLAM systems which groups data points into 2D lines, because this moving-field-of-view issue applies there too.

We propose to use different criteria for associating View segments to Map ones. The reason for the change is the problem of moving-field-of-view of the sensing system, which turns in a moving window on the world feature, see Figure 5. Our proposal is of interest also for 2D-3DoF SLAM systems which groups data points into 2D lines, because this moving-field-of-view issue applies there too.

5.1 Criterium $N°1$

The first criterium we use is the Mahalanobis distance between the support line of the map segment and the extrema of the view segment (see figure 6).

The support line of segment m_j is:

$$\left\{ \begin{array}{l} x = x_j' + (x_j'' - x_j')t \\ y = y_j' + (y_j'' - y_j')t \\ z = z_j' + (z_j'' - z_j')t \end{array} \right. \quad (10)$$

where $[x_j', y_j', z_j']^T$ are the coordinate of an extremum of segment m_j and $[x_j'', y_j'', z_j'']^T$ are the coordinates of

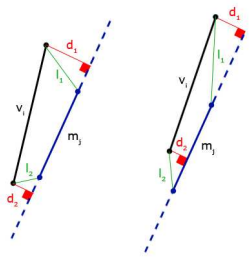

Figure 6: Comparison between point-to-point distance (green line) and point-to-supportline distance (red line). Notice the difference between d_i and l_i when the moving-window problem happens

the other. The distance of a point $\mathbf{P}=[x_i, y_i, z_i]^T$, e.g. an extremum of a v_i segment, from the line is:

$$D(\mathbf{P}, m_j) = \sqrt{\frac{A^2 + B^2 + C^2}{(x_j'' - x_j')^2 + (y_j'' - y_j')^2 + (z_j'' - z_j')^2}}$$

being:

$$A = det \begin{bmatrix} x_i - x_j' & y_i - y_j' \\ x_j'' - x_j' & y_j'' - y_j' \end{bmatrix} \quad (11)$$

$$B = det \begin{bmatrix} x_i - x_j' & z_i - z_j' \\ x_j'' - x_j' & z_j'' - z_j' \end{bmatrix} \quad (12)$$

$$C = det \begin{bmatrix} y_i - y_j' & z_i - z_j' \\ y_j'' - y_j' & z_j'' - z_j' \end{bmatrix} \quad (13)$$

We have to deal with the two extrema of segment v_i, therefore the considered distance is:

$$\mathbf{h}(v_i, m_j) = \begin{bmatrix} D(v_i', m_j) & D(v_i'', m_j) \end{bmatrix}^T \quad (14)$$

It is now possible to compute the Mahalanobis distance between the two segments and compare it with the threshold:

$$D_{\mathbf{h}}^2 = \mathbf{h^T C_h^{-1} h} < \chi_{d,\alpha}^2 \quad (15)$$

The covariance $\mathbf{C_h}$ of $\mathbf{h}$ is obtained as:

$$\mathbf{C_h} = \mathbf{H P}_{m_j} \mathbf{H^T} + \mathbf{G P}_{v_i} \mathbf{G^T} \quad (16)$$

where:

$$\mathbf{H_{(2,6)}} = \mathbf{J} \frac{\mathbf{h}}{\mathcal{M}} = \begin{bmatrix} \frac{\partial D(vw_i', m_j)}{\partial \mathcal{M}} & \frac{\partial D(v_i'', m_j)}{\partial \mathcal{M}} \end{bmatrix}^{\mathbf{T}} \quad (17)$$

$$\mathbf{G_{(2,6)}} = \mathbf{J} \frac{\mathbf{h}}{\mathcal{V}} = \begin{bmatrix} \frac{\partial D(v_i', m_j)}{\partial \mathcal{V}} & \frac{\partial D(v_i'', m_j)}{\partial \mathcal{V}} \end{bmatrix}^{\mathbf{T}} \quad (18)$$

5.2 Criterium $N°2$

The second criterium we use considers the angle between the support lines of the two segments; this constraint complements the previous in avoiding the association of a short segment to another one perpendicular to it. We can write the two support lines as:

$$\begin{cases} x = a_1 t + x_i' & a_1 = (x_i'' - x_i') \\ y = b_1 t + y_i' & b_1 = (y_i'' - y_i') \\ z = c_1 t + z_i' & c_1 = (z_i'' - z_i') \end{cases} \quad (19)$$

and

$$\begin{cases} x = a_2 t + x_j' & a_2 = (x_j'' - x_j') \\ y = b_2 t + y_j' & b_2 = (y_j'' - y_j') \\ z = c_2 t + z_j' & c_2 = (z_j'' - z_j') \end{cases} \quad (20)$$

the angle between them is:

$$angle_{ji} = cos^{-1} \left(\frac{a_1 a_2 + b_1 b_2 + c_1 c_2}{\sqrt{a_1^2 + b_1^2 + c_1^2} \sqrt{a_2^2 + b_2^2 + c_2^2}} \right) \quad (21)$$

The covariance is:

$$\mathbf{C_a} = \mathbf{F P}_{m_j} \mathbf{F^T} + \mathbf{G P}_{v_i} \mathbf{G^T} \quad (22)$$

$$\mathbf{F_{(1,6)}} = \mathbf{J} \frac{angle_{ji}}{\mathcal{M}} = \begin{bmatrix} \frac{\partial angle_{ji}}{\partial \mathcal{M}} \end{bmatrix} \quad (23)$$

$$\mathbf{G_{(1,6)}} = \mathbf{J} \frac{angle_{ji}}{\mathcal{V}} = \begin{bmatrix} \frac{\partial angle_{ji}}{\partial \mathcal{V}} \end{bmatrix} \quad (24)$$

and the constraint is:

$$D_{ji}^2 = angle_{ji} \mathbf{C_a^{-1}} angle_{ji} < \chi_{d,\alpha}^2 \quad (25)$$

5.3 Criterium $N°3$

The last criterium is not a probabilistic one; instead it is related to the projections of the segment extrema on the other segment; one of the following two conditions has to hold to get an association:

1. at least one of the extrema of the view segment has to project onto the map segment;

2. both the extrema of the view segment project outside the map segment, but the extrema of the map segment project inside the view segment.

5.4 Multi-Criteria Nearest Neighbor

In this section we present a variation on the nearest neighbor data association algorithm that uses the classic ICP (Iterative Closest Point) algorithm [4] and the multi-criteria approach. In the following we compare the performance of this approach and the joint compatibility branch and bound.

1. take v_i from $\mathcal{V}$ and move it into the Map reference frame
2. compute a match measure for all couples $< v_i, m_j > (\forall m \in \mathcal{M})$ (the two distances mentioned above for 1st and 2nd criterium); if the 3rd criterium does not result in some intersection between v_i and m_j, then this match is not possible
3. order the list of matches $< v_i, m_j >$ in decreasing order of match measure (first respect to the 1st criterium and then respect to the 2d criterium), for all $m_j s$
4. repeat the above for all $\mathcal{V}$ segments $(\forall v_i \in \mathcal{V})$
5. order all rows related to v_i according to the same match measure
6. select as a match the couple $< v_i, m_j >$ with the least match measure, if the match measure is lower than a given threshold
7. erase any other couple including v_i or m_j from the 2D list
8. if some head-of-row is erased, reorder the rows
9. repeat from the selection of the best match
10. basing on the matches found so far, compute the robot pose
11. restart at where the view segments are moved into the map
12. continue repeating the loop above until reaching a steady state in the pose (and associations)

5.5 Multi-Criteria Joint Compatibility

In this section we integrate the three criteria in the joint compatibility branch and bound technique. As we have described above, we need to generate an hypothesis H of correspondences. The exploration of the interpretation tree is like in [10]. Suppose we have two sets of features: $\mathcal{V}$ and $\mathcal{M}$ to associate. One path in the tree from the root to the node i represents a hypothesis $H_k = \{< v_1, m_{j_1} >, ..., < v_i, m_{j_i} >\}$. To establish the correctness of this hypothesis we use the following joint compatibility function $\mathbf{f}_{H_k}$:

$$\mathbf{f}_{H_k} = \begin{pmatrix} D(v'_1, m_{j_1}) \\ D(v''_1, m_{j_1}) \\ \vdots \\ D(v'_i, m_{j_i}) \\ D(v''_i, m_{j_i}) \end{pmatrix}$$

being D the distance between the support line of the map segment and the extrema of the view segment.

Before to evaluate this function, to reduce the amount of computational time required to explore one branch, we compute the 3rd criterium on the couple $< v_i, m_{j_i} >$ because it is very selective and fast. Subsequently we compute the 2nd criterium on the same couple, to decrease the possibility to compute in vain the next step. Finally, we use the 1st criteria in the joint compatibility test.

6 Experiments on Real Data

We implemented a known SLAM algorithm (Hierarchical SLAM [11],[6]), but on 3D segments (see [9]). The algorithm is based on EKF, Mahalanobis distance, hierarchical decomposition of the data structure, and interpretation trees. This approach is known since many years and have been used with many sensing systems (sonar, laser range finders, etc.); it has been used also with a 3D vision system like the one we used, but with data obtained by projecting on the floor the 3D data [10]. The key point in Hierarchical SLAM is the reduction in complexity which is attained by limiting (by means of the hierarchical decomposition of the data structure) the number of items involved in the most cumbersome SLAM phases. In particular it bases on global and local world models. The two represent two abstraction levels in considering the space. At the local level we have the submap, a limited cardinality set of world features. The global level is a graph-based representation of the world, where the submaps are the nodes and the edges the (pose) constraints between nodes. Another relevant aspect of this approach concern loop-closing for large-sized loops, an issue which has not been presented in this work (if interested see [9]).

For the experimental activity we used a mobile robot from Robosoft which computes odometry as a 3DoF pose; this datum reaches a PC via serial line. On the PC we have a frame grabber capable to grab three 704x558x8 pixels images at the same time. Each channel of the frame grabber is connected to a Sony XC75CE camera. Cameras have been calibrated with

a standard DLT approach like in [5]. The robot has been moved, by hand due to a servo-amplifier failure, inside the 4th floor of building U7, Univ. Milano - Bicocca, Milano, Italy. Distances between consecutive robot poses, i.e. views, was about 0.05m. The overall distance travelled has been about 200m (1900 views - average of 20 segments for view). The map in Figure 6-7 is built using all these views.

Table 1 shows the comparison between two data-association methods on the same data: one using the joint compatibility branch and bound (robust state-of-the-art data association in SLAM) with segment extrema (state-of-the-art when we dealing with segments) and the other using the JCBB with our multi-criteria. This comparison bases on counting the number of matches obtained during the SLAM activity (but before any loop-closure). A strong increment is perceivable in the number of associations, with respect to the method based on segment extrema. This is caused by the of moving-field-of-view problem mentioned above, which is well handled by our approach.

Table 2 presents the number of matches using the JCBB with multi-criteria versus the number of matches when using the multi-criteria nearest neighbor algorithm (based on ICP method). Here the increment w.r.t. joint compatibility is much less, because of the use of multi-criteria and ICP algorithm.

Figure 7 shows the data-association errors: on the left 2 segments reconstructed using multi-criteria JCBB, on the right the same segments using JCBB with segment extrema. Notice that the segments on the right are short segments and not matched with the long one: the distance between extrema is too high. In the left segments are correctly associated.

Figure 8 and 9 show the comparison between a section of the map reconstructed with JCBB with segment extrema method and the same section generated by the multi-criteria JCBB. The two maps are reconstructed using Hierarchical SLAM approach. We notice that the former have more (noisy) segments, which did not correctly match (missed matches) especially in the robot motion direction (long segments affected by the moving-window problem), while in the latter we have less un-matched segments.

Finally, in Figure 10 we present the relative error (difference between errors) on the robot pose during the exploration when the map is built using multi-criteria JCBB, JCBB with segment extrema and multi-criteria ICP. This graph shows the difference between the areas of the uncertainly ellipses (σ) generated for the robot poses. The pink line is the difference between the multi-criteria JCBB and the JCBB segment extrema method. Notice that the relative error is negative, therefore the multi-criteria JCBB produces less robot errors and better matches w.r.t. JCBB with segment extrema. The second graph (blue line) is the difference between the multi-criteria JCBB and multi-criteria ICP. We still have better matches using multi-criteria JCBB.

JCBB Extrema	Multi-criteria JCBB	Incr.%
1839	2783	51%

Table 1: Comparison between the number of data-associations with JCBB segment extrema method and multi-criteria JCBB

Multi-criteria ICP	Multi-criteria JCBB	Incr.%
2489	2783	11%

Table 2: Comparison between the number of data-associations with multi-criteria ICP and multi-criteria JCBB

7 Conclusion

We presented work dealing with the SLAM problem when the data sensed are 3D. Vision systems generating 3D data are quite reliable and used since many years in the robotic community, but SLAM with 3D vision is not common. The work deals with data association issues and the related accuracy of estimates when the sensed data are 3D segments. Our claims are: 1) robot is immersed in 3D environment and using redundant sensing systems is expansive; 2) the use of different data-association criteria improves the search and turns into less data association errors; 3) reliable 3D world mapping can be obtained in large indoor environments using our proposed improvements.

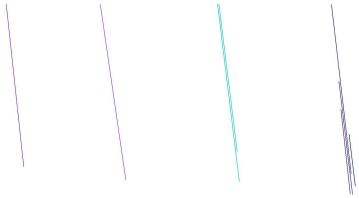

Figure 7: On the left two correctly reconstructed segments in one view; on the right the same view with short segments not associated to the long one

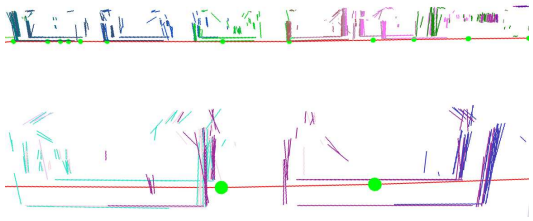

Figure 8: Submaps generated by using JCBB segment extrema method. In foreground part of the corridor, in background another corridor (some 30 m farther). Notice the presence of many missed matches.

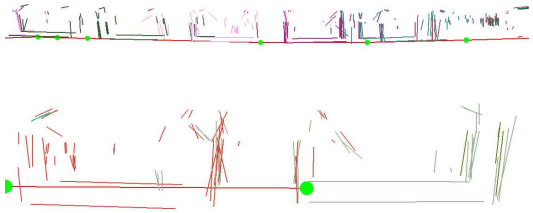

Figure 9: Submaps as above but generated with the multi-criteria JCBB. Notice that the number of unmatched segments, much less than for the JCBB segment extrema

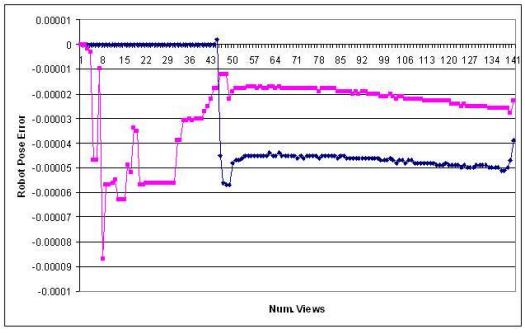

Figure 10: Relative robot pose error during the reconstruction of 144 views. The graph represents the difference with respect to the robot pose updated using the multi-criteria JCBB (taken as the best): the pink line difference from JCBB segment extrema, the blue line difference from multi-criteria ICP

References

[1] Colin Angle. Invited talk at the annual euron meeting, amsterdam. Author is CEO of IS-Robotics, March 2004.

[2] N. Ayache and F. Lustman. Trinocular stereo vision for robotics. *IEEE Trans. on PAMI*, 12(1), 1991.

[3] N. J. Ayache and O. D. Faugeras. Maintaining representations of the environment of a mobile robot. *IEEE Trans. on R&A*, 5(6):804–819, 1989.

[4] Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. 14:239–256, February 1992.

[5] O. Faugeras e G. Toscani. The calibration problem for stereo. In *Proc. IEEE CVPR*, pages 15–20, 1986.

[6] C. Estrada, J. Neira, and J. D. Tardós. Hierarchical slam: real-time accurate mapping of large environments. *IEEE Trans. on Robotics*. to appear.

[7] P. Kahn, L. Kitchen, and E. M. Riseman. A fast line finder for vision-guided robot navigation. *IEEE Trans. on PAMI*, 12(11), Nov 1990.

[8] Kurt Konolige. Small vision system. hardware and implementation. Hayama, Japan, 1997.

[9] Daniele Marzorati, Domenico G. Sorrenti, and Matteo Matteucci. 3d-6dof hierarchical slam with trinocular vision. In *Proc. of ECMR Conf.*, 2005.

[10] J. Neira and J. D. Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Trans. R&A*, 17(6):890–897, 2001.

[11] J. Neira, J. D. Tardós, and J. A. Castellanos. Linear time vehicle relocation in slam. In *Proc. IEEE ICRA*, volume 1, pages 427–433, Sept 2003.

[12] A. Rottmann, O. Martínez Mozos, C. Stachniss, and W. Burgard. Place classification of indoor environments with mobile robots using boosting. Pittsburgh, PA, USA, 2005.

[13] H. Surmann, A. Nüchter, K. Lingemann, and J. Hertzberg. 6d slam: Preliminary report on closing the loop in six dimensions. In *5th IFAC Symp. IAV*, June 2004.

[14] Oliver Wulf, Bernardo Wagner, and Mohamed Khalaf-Allah. Using 3d data for monte carlo localization in complex indoor environments. In *Proc. of ECMR Conf.*, 2005.