6DOF SLAM with Stereo-in-hand

L. M. Paz, P. Piniés, J.D. Tardós, J Neira

Abstract— **In this paper we describe a system that carries out SLAM using a stereo pair moving with 6DOF as the only sensor. Textured point features are extracted from the images and stored as 3D points if seen in both images with sufficient disparity, or stored as inverse 3D points otherwise. This allows the system to make use of both near and far features that provide distance and orientation, or orientation information, respectively. Unlike other vision only SLAM systems, stereo does not suffer from 'scale drift' because of unobservability problems, and thus requires no other information such as gyroscopes or accelerometers. Our SLAM algorithm generates sequences of conditionally independent local maps that can share information related to the camera motion and common features being tracked. The system computes the full map using the Divide and Conquer algorithm adapted for conditionally independent local maps, allowing linear time execution. We show experimental results in outdoor urban environments that demonstrate the robustness and scalability of our system.**

I. INTRODUCTION

The interest in using cameras in SLAM has grown tremendously in recent times. Cameras have become much more inexpensive than lasers, and also provide texture rich information about scene elements at practically any distance from the camera. In applications where it is not practical to carry heavy and bulky sensors, such as egomotion for people tracking and environment modelling in rescue operations, cameras are light weight sensors that can be easily adapted to helmets used by rescuers, or simply worn.

Currently, visual SLAM systems have been demonstrated to be viable for small environments, such as MonoSLAM [1] as well as moderately large ones, such as Hierarchical Visual SLAM [2]. A single camera is used in both these systems, and thus scale unboservability is a fundamental limitation in both. Either the scale is fixed by observing a known object as is usually done in MonoSLAM, or drift in scale can occur, as is reported in the Hierarchical Visual SLAM system. Furthermore, MonoSLAM is an EKF SLAM system, and cannot be used to map large environments. Hierarchical Visual SLAM can be used for large scale mapping because the system works on local maps of limited size, achieving constant time execution most of the time. These local maps are organized in a hierarchical structure, with an adjacency graph at the upper level where the relative transformation between consecutive or loop closing local maps is maintained.

Another important requirement of the Hierarchical Visual SLAM system is that local maps be statistically independent, so that the adjacency graph can be efficiently maintained. This is achieved by creating a new local map from scratch every time the current local map size limit has been reached. This impedes sharing valuable information between local maps, such as the camera velocity, or information about features being currently observed. A recent SLAM algorithm also based on local mapping and of linear time execution, Divide and Conquer SLAM [3], can only accommodate independent local maps in its current version.

Since the initial results of [4] great progress has been made in the related problem of visual odometry [5], [6]. Visual odometry systems are however incapable of closing loops, and thus eventual drift is inevitable. We consider this one of the main advantages of SLAM, and thus are interested in developing a real time, low cost robust visual SLAM system.

In this paper we propose a visual SLAM system with the following two main highlights:

1) Unlike any other visual SLAM system, we consider information from features both close and far from the cameras. Stereo provides 3D information from nearby scene points, and each camera also provides angular information from distant scene points. Both types of information are incorporated into the map and used to improve the estimation of both the camera pose and velocity, as well as the map. Nearby scene points also provide scale information through the stereo baseline, eliminating the scale unobservability problem.
2) We use Conditionally Independent Divide and Conquer SLAM, a novel combination of conditionally independent local maps [7] and the Divide and Conquer SLAM algorithm that allows to maintain both camera velocity information and current feature information during local map initialization. This adds robustness to the system without sacrificing precision or consistency in any way. It also allows linear time execution, enabling the system to be used for large scale indoor/outdoor SLAM.

This paper is organized as follows: the general structure of the system and the feature detection process are described in section II. In section III we detail the process of building conditionally independent local maps. In section IV we describe the Conditionally Independent Divide and Conquer SLAM algorithm. In section V we detail two experiments carried out to test the system: and indoor 220m loop and an outdoor 140m loop. Section VI contains our conclusions and future work.

L. M. Paz, P. Piniés, J.D. Tardós, J Neira are with the Departamento de Informática e Ingenieria de Sistemas, Centro Politécnico Superior, Universidad de Zaragoza, Zaragoza, Spain {linapaz, ppinies, tardos, jneira}@unizar.es

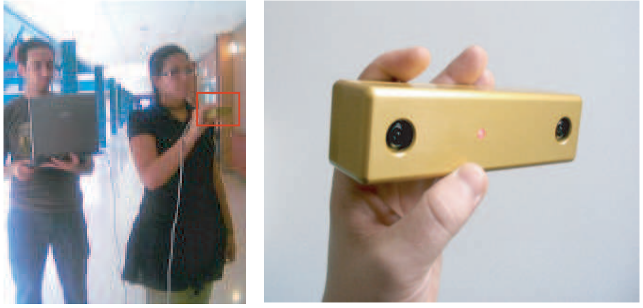

Fig. 1. Bumblebee Stereo vision system used to acquire images sequences. Picture on the left shows the experimental setup during the data acquisition for the indoor experiment.

II. Detection and Tracking of 3D points and Inverse 3D points

Our 6DOF system consists of a stereo camera carried in hand and a laptop to record and process a sequence of images, Fig. 1. As it is known, a stereo camera can provide depth estimation of points up to a certain distance determined by the baseline between left and right cameras. Therefore, two regions can be identified: one close to the camera in which it behaves as a range and bearing sensor, and the other in which the stereo becomes a monocular camera, only providing bearing measurements of points. To take advantage of both types of information, we combine depth points and inverse 3D points in the state vector in order to build a map and estimate the camera trajectory.

During the estimation process right image is chosen as reference to initialize new features. Interesting points are extracted from it and classified according to their disparity with the left image. Those points whose disparity reveals a close distance are initialized as 3D features, otherwise they are modelled as inverse depth points and initialized using the bearing information obtained from the right image. When the camera moves, these features are tracked in order to update the filter and produce the corresponding corrections. To track a feature, its position is searched in both images inside a bounded region given by the uncertainty in the camera motion and the corresponding uncertainty of the feature.

The algorithm to select, initialize and manage these features is explained in the following subsections.

A. Selection and Management of Trackable points

To ensure tracking stability of map features, distinctive points have to be selected. Following a similar idea presented in [8], we use Shi-Tomasi variation of Harris corner detector to select good trackable image points and the associated 11x11 surrounding patch to perform correlation when solving data association.

From the first step, the right image is split up in regular buckets so that the point with the best detector response per cell is selected, see Fig. 2 top. This structure is maintained during the following steps to add features when we found empty cells, which allow us to distribute features uniformly in the image. The approach is accompanied by a feature management strategy, so that non-persistent features are deleted from the state vector to avoid an unnecessary growth in population.

B. 3D points Initialization

Corners classified as depth points are transformed to 3D points given the disparity information that comes from the stereo pair. We take advantage of the stereo camera capability to provide rectify images. Then the back-projection equations to obtain the 3D point correspond to a pinhole camera model. Consider Eq.(1)

$$\begin{aligned} d &= u_l - u_r \\ x &= \frac{b(u_r - u_0)}{d} \\ y &= \frac{b(v_r - v_0)}{d} \\ z &= \frac{fb}{d} \end{aligned} \tag{1}$$

this equation relates images points and 3D points using the transformation function $\mathbf{x}_{3d} = f(u_r, v_r, u_l, v_l) = (x, y, z)^T$, where (u_r, v_r) and (u_l, v_l) are the pixels on the right and left images with an associated pixel uncertainty, and d is the horizontal disparity. The remainder terms in the equations are the calibrated parameters of the camera where (u_0, v_0) is the central pixel of the images, b is the baseline and f is the focal length.

C. Inverse points Initialization

Current research on Monocular SLAM has shown that the inverse-depth points parametrization introduced in [9] is suitable to represent the distribution of the features at the infinity as well as perform undelayed initialization. Given the camera location

$$\mathbf{x}_c = \begin{bmatrix} \mathbf{r}_c \\ \Theta_c \end{bmatrix} \tag{2}$$

from which the feature in the image was first observed, an inverse-depth point can be defined as

$$\mathbf{x}_{inv} = \begin{bmatrix} \mathbf{r}_c \\ \theta \\ \phi \\ 1/\rho \end{bmatrix} \tag{3}$$

This vector depends on the optical center pose $\mathbf{r}_c$, the orientation of the ray passing through the image point (i.e. azimuth θ, elevation ϕ), and the inverse of its depth, ρ.

D. Predicting observations in a stereo pair

In each step, all features in the camera field of view have to be projected on the current stereo pair in order to carry out an active search. The prediction equation for Inverse-depth features can be extended to represent its projective rays for

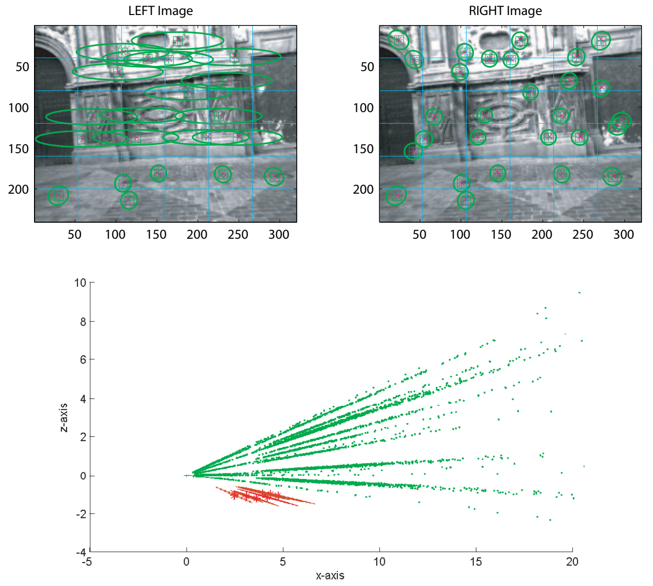

Fig. 2. Points detected using a stereo camera. Projection of map features (top): 3D features projections yield more precise search regions on both images; inverse-depth produces larger regions on the left image in which active search is perform. We show features uncertainties from a lateral perspective (bottom): 3D features uncertainties are drawn using red ellipses whereas we use samples to show the Inverse-depth features uncertainties.

the left and right cameras:

$$\begin{aligned} \mathbf{y}^R_{r_{inv}} &= h_{r_{inv}}(\mathbf{x}_{BR}, \mathbf{x}^B_{inv}) \\ &= Rot(\Theta_{BR})^T((\mathbf{r}^B_c - \mathbf{r}^B_{BR}) + \frac{1}{\rho}\mathbf{m}^B) \\ \mathbf{y}^R_{l_{inv}} &= h_{l_{inv}}(\mathbf{x}_{BR}, \mathbf{x}^B_{inv}) \\ &= Rot(\Theta_{BR})^T((\mathbf{r}^B_c - \mathbf{r}^B_{BR}) + \frac{1}{\rho}\mathbf{m}^B) - \mathbf{x}^R_{rl} \end{aligned} \tag{4}$$

These equations transform the inverse-depth feature $\mathbf{x}^B_{inv}$ expressed in the base map reference B to a 3D point $\mathbf{y}^R_{inv}$ in the current camera reference R being $\mathbf{m}^B$ the unitary ray directional vector:

$$\mathbf{m}(\theta, \phi) = \begin{bmatrix} \cos(\theta)\cos(\phi) \\ \sin(\theta)\cos(\phi) \\ \sin(\phi) \end{bmatrix} \tag{5}$$

Vector $\mathbf{x}^R_{rl} = [0 \ b \ 0]^T$ represents the rigid transformation in the camera frame between left and right cameras, which only depends on the baseline.

In a similar way, we describe observations corresponding to 3D map features, for which we use composition operators:

$$\begin{aligned} \mathbf{y}^R_{r_{3d}} &= h_{r_{3d}}(\mathbf{x}_{BR}, \mathbf{x}^B_{3d}) \\ &= \ominus\mathbf{x}_{BR} \oplus \mathbf{x}^B_{3d} \\ \mathbf{y}^R_{l_{3d}} &= h_{l_{3d}}(\mathbf{x}_{BR}, \mathbf{x}^B_{3d}) \\ &= \ominus\mathbf{x}_{BR} \oplus \mathbf{x}^B_{3d} - \mathbf{x}^R_{rl} \end{aligned} \tag{6}$$

Fig. 2 top shows the prediction of those 3D and inverse-depth features that falls inside the field of view of both images. Some interesting characteristics can be pointed out when working with a stereo camera given that features projected in both images can produce a pair of independent observations. For instance, first time an inverse-depth feature is seen, larger search regions are produced on the left image due to the uncertainty projection. However, when the feature is updated in the next step, the uncertainty is drastically reduced which means that features can not be in the stereo nearby region.

When our system starts moving, features projection may disappear from the field of view of one camera. Nevertheless, information to update the state is still available if the feature is projected in the other camera.

III. Generating Conditionally Independent Local Maps

Our implementation of the SLAM algorithm is based on local map techniques since they provide good consistency properties and low computational requirements during the estimation. In addition, we are interested in sharing some state vector components between consecutive submaps. Otherwise, some camera states, such as linear and angular velocities, that have been estimated in a map should be suddenly discarded when a new submap is initiated. Furthermore, features that are near to the transition region between adjacent submaps can be shared as well in order to improve local maps relative location. Nevertheless, special care is needed if both submaps are joined in a single map since their estimates are not independent anymore.

The novel technique to achieve these requirements is based on the Conditionally Independent Local Maps CI [7]. To ease the explanation of the algorithm we will give a brief review of the technique before going to the particular details of the actual method.

A. Brief Review of Conditionally Independent Local Maps

Suppose that a local map 1 has been built and we want to start a new submap 2 but sharing some elements in common with 1. Submap 1 is described by the following probability density function:

$$p(\mathbf{x}_A, \mathbf{x}_C | \mathbf{z}_a) = \mathcal{N}\left(\begin{bmatrix} \hat{\mathbf{x}}_{A_a} \\ \hat{\mathbf{x}}_{C_a} \end{bmatrix}, \begin{bmatrix} P_{A_a} & P_{AC_a} \\ P_{CA_a} & P_{C_a} \end{bmatrix} \right) \tag{7}$$

where $\mathbf{x}_A$ are the components of the current submap that only belong to 1, $\mathbf{x}_C$ are the elements that will be shared with 2 and $\mathbf{z}_a$ the observations gathered during the map construction. Notice that upper case subindexes are for state vector components whereas lower case subindexes describe which observations $\mathbf{z}$ have been used to obtain the estimate.

Submap 2 is then started with the result of marginalizing out the non common elements from 1.

$$p(\mathbf{x}_C | \mathbf{z}_a) = \int p(\mathbf{x}_A, \mathbf{x}_C | \mathbf{z}_a)\, d\mathbf{x}_A = \mathcal{N}(\hat{\mathbf{x}}_{C_a}, P_{C_a}) \tag{8}$$

During the trajectory along map 2 new observations $\mathbf{z}_b$ are gathered from previous components $\mathbf{x}_C$ as well as observations of new elements $\mathbf{x}_B$ that are incorporated to the

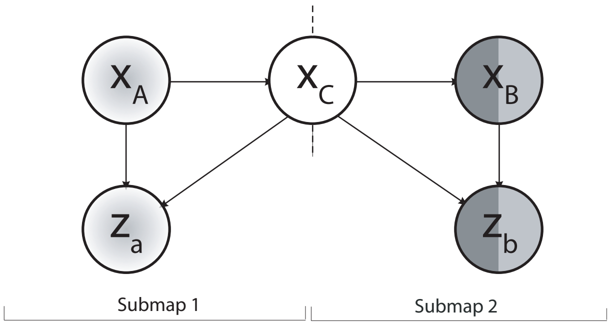

Fig. 3. Bayesian net that describes the relations between consecutive submaps

map. When map 2 is finished, its estimate is finally described by:

$$p(\mathbf{x}_C, \mathbf{x}_B | \mathbf{z}_a, \mathbf{z}_b) = \mathcal{N}\left(\begin{bmatrix} \hat{\mathbf{x}}_{C_{ab}} \\ \hat{\mathbf{x}}_{B_{ab}} \end{bmatrix}, \begin{bmatrix} P_{C_{ab}} & P_{CB_{ab}} \\ P_{BC_{ab}} & P_{B_{ab}} \end{bmatrix} \right) \tag{9}$$

where the subindexes in the estimates $\hat{\mathbf{x}}_{C_{ab}}$ and $\hat{\mathbf{x}}_{B_{ab}}$ reveal that both sets of observations $\mathbf{z}_a$ and $\mathbf{z}_b$ have been used in the estimation process. This means that submap 2 is updated with all the information gathered by the sensor. Recall that map 1 in Eq(7) has been updated with the observation $\mathbf{z}_a$ but not with the more recent observation $\mathbf{z}_b$.

Figure(3) shows a Bayesian network that describes the probabilistic dependencies between elements of submaps 1 and 2. As can be seen, the only connection between the set of nodes ($\mathbf{x}_A$, $\mathbf{z}_a$) and ($\mathbf{x}_B$, $\mathbf{z}_b$) is through node $\mathbf{x}_C$, i.e. both subgraphs are *d-separated* given $\mathbf{x}_C$ [10]. This implies that nodes $\mathbf{x}_A$ and $\mathbf{z}_a$ are *Conditionally Independent* of nodes $\mathbf{x}_B$ and $\mathbf{z}_b$ given node $\mathbf{x}_C$. Intuitively this means that if $\mathbf{x}_C$ is known, submaps 1 and 2 do not carry any additional information about each other.

Taking this structure into account, it can be demonstrated that the influence of the new observations $\mathbf{z}_b$ in the $\mathbf{x}_A$ components of submap 1 can be *back-propagated* using the following equations:

$$\begin{aligned} K &= P_{AC_a} P_{C_a}^{-1} \\ &= P_{AC_{ab}} P_{C_{ab}}^{-1} && (10) \\ P_{AC_{ab}} &= K P_{C_{ab}} && (11) \\ P_{A_{ab}} &= P_{A_a} + K(P_{CA_{ab}} - P_{CA_a}) && (12) \\ \hat{\mathbf{x}}_{A_{ab}} &= \hat{\mathbf{x}}_{A_a} + K(\hat{\mathbf{x}}_{C_{ab}} - \hat{\mathbf{x}}_{C_a}) && (13) \end{aligned}$$

Therefore, using this technique we can independently build local maps that have elements in common and afterwards retrieve the global information in a consistent manner. The process to join several local maps in a single state will be explained in section IV.

B. Actual implementation for the stereo

We now explain how the technique of conditionally independent local maps is applied in our implementation of the stereo system.

Since the camera moves in 6DOF, the camera state is composed of its position using cartesian coordinates, the orientation in euler angles and its linear and angular velocities. As we mentioned before, 3D points and inverse depth points are included as features in the state vector. When a local map $\mathbf{m}_i$ is finished, the final map estimate is given by:

$$\mathbf{m}_i.\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{x}}_{R_iR_j} \\ \hat{\mathbf{v}}_{R_iR_j} \\ \hat{\mathbf{x}}_{R_iF_{1:m}} \\ \hat{\mathbf{x}}_{R_iF_{m+1:n}} \end{bmatrix} \tag{14}$$

where $\hat{\mathbf{x}}_{R_iR_j}$ is the camera location in local reference coordinates, $\hat{\mathbf{v}}_{R_iR_j}$ are the linear and angular velocities, $\hat{\mathbf{x}}_{R_iF_{1:m}}$ are 3D and inverse depth features that will only remain in the current map and $\hat{\mathbf{x}}_{R_iF_{m+1:n}}$ are 3D and inverse depth features that will be shared with the next submap $\mathbf{m}_j$.

Since the current camera velocity $\hat{\mathbf{v}}_{R_iR_j}$ and some features $\hat{\mathbf{x}}_{R_iF_{m+1:n}}$ are used to initialize the next local map, a local copy of these elements have to be calculated and added to the current submap.

$$\mathbf{m}_i.\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{x}}_{R_iR_j} \\ \hat{\mathbf{v}}_{R_iR_j} \\ \hat{\mathbf{x}}_{R_iF_{1:m}} \\ \hat{\mathbf{x}}_{R_iF_{m+1:n}} \\ \cdots \\ \ominus\hat{\mathbf{x}}_{R_iR_j} \oplus \hat{\mathbf{v}}_{R_iR_j} \\ \ominus\hat{\mathbf{x}}_{R_iR_j} \oplus \hat{\mathbf{x}}_{R_iF_{m+1:n}} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}}_{A_a} \\ \cdots \\ \hat{\mathbf{x}}_{C_a} \end{bmatrix} \tag{15}$$

where the new elements define the common part $\hat{\mathbf{x}}_{C_a}$ and the original map defines $\hat{\mathbf{x}}_{A_a}$. Notice that the appropriate composition operation have to be applied for each transformed component and that the corresponding covariance elements have to be added to the map.

In local mapping, a reference have to be identified to start a new map. This common reference is represented by the final vehicle position, which is the case of R_j between $\mathbf{m}_i$ and $\mathbf{m}_j$.

The initial state vector of the next submap is then given by:

$$\mathbf{m}_j.\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{x}}_{R_jR_j} \\ \ominus\hat{\mathbf{x}}_{R_iR_j} \oplus \hat{\mathbf{v}}_{R_iR_j} \\ \ominus\hat{\mathbf{x}}_{R_iR_j} \oplus \hat{\mathbf{v}}_{R_iR_j} \\ \ominus\hat{\mathbf{x}}_{R_iR_j} \oplus \hat{\mathbf{x}}_{R_iF_{m+1:n}} \end{bmatrix} \tag{16}$$

where $\hat{\mathbf{x}}_{R_jR_j}$ represents the location of the camera in the new reference frame with zero uncertainty and zero correlation with the rest of the elements of the initial map. Notice that the initial velocity brought from the previous map has been replicated twice. Since it is a dynamical quantity one of the copies will change as the camera moves through the new map carrying the current camera velocity. The other copy will remain fixed and used with the transformed features as a common element to *backpropagate* the information adequately. The same process is successively repeated with all local maps.

IV. Conditionally Independent Divide and Conquer SLAM

Divide and Conquer SLAM (D&C) has proved to be a good algorithm to join local maps minimizing the computational complexity of EKF-based SLAM and improving consistency. The algorithm allows us to join efficiently several local maps in a single state vector using Map Joining in a Hierarchical tree structure [11].

We have adapted D&C SLAM to work with conditional independent local maps by redefining the Map Joining process. Consider two CI local maps that belong to the same level of the tree. The resulting map after the join will be defined by:

$$\begin{aligned} & p(\mathbf{x}_A, \mathbf{x}_B, \mathbf{x}_C | \mathbf{z}_a, \mathbf{z}_b) = \\ & = \mathcal{N}\left(\begin{bmatrix} \hat{\mathbf{x}}_{A_{ab}} \\ \hat{\mathbf{x}}_{C_{ab}} \\ \hat{\mathbf{x}}_{B_{ab}} \end{bmatrix}, \begin{bmatrix} P_{A_{ab}} & P_{AC_{ab}} & P_{AB_{ab}} \\ P_{CA_{ab}} & P_{C_{ab}} & P_{CB_{ab}} \\ P_{BA_{ab}} & P_{BC_{ab}} & P_{B_{ab}} \end{bmatrix} \right) \end{aligned} \tag{17}$$

Therefore, to recover the full map from Eqs. (7) and (9), we have to apply the following steps:

- The first map has to be updated with the new observations obtained in the second map using equations (10), (11), (12) and (13).
- The correlations terms between the non-common elements of both maps are computed with the next equation:

$$\begin{aligned} P_{AB_{ab}} &= P_{AC_{ab}} P_{C_{ab}}^{-1} P_{CB_{ab}} \\ &= K P_{CB_{ab}} \end{aligned} \tag{18}$$

  which can be obtained taking into account the structure of the CI maps [7].
- Replication of common elements are deleted in both maps.
- Elements belonging to the second map are transformed to the first map reference.

A. Data association for D&C SLAM

As it was seen in the previous sections (III, IV), we build CI local maps in the spirit of a EKF-based SLAM and perform D&C SLAM algorithm to carry out the map fusion task. Given this outline we will consider the data association problem from the conventional point of view when a local map is built, and from the Map joining perspective.

1) Data association for CI local maps: Recent work on large environments [2] has shown that Joint Compatibility avoids map corruption by rejecting measurements that come from moving objects. This framework turns out to be suitable in environments with a few number of observations. However, even though impressive results were registered, a Branch and Bound algorithm implementation (**JCBB**) limits its use when the number of observations per step increases. In this paper we have obtained more efficient results using the *Randomized Joint Compatibility* version **RJC** proposed in [11], in which a *joint compatibility* **JC** test is run with a fixed set of measurements p selected randomly. In this case correlation between patches and Individual compatibility tests have been used as in the previous work to obtain candidate matches. If all p measurements and its matches are compatible, we apply the Nearest Neighbor rule to match the remaining measurements. Once a total hypothesis H is obtained, we check **JC** to avoid false positives. The process is repeated t times in the spirit of an adaptive RANSAC limiting the probability of missing a correct association.

2) Data association for the D&C Map Joining process: The property of sharing common elements solves the data association problem between consecutive local maps [11]. This leads us to solve data association just in loop closing situations. We use Maximum Clique Algorithm in order to detect an already visited area [2]. The algorithm finds correspondences between features in different local maps, taking into account the texture and the relative geometry among features. Once corresponding features are found, an ideal measurement equation that imposes the loop closing constraint is applied.

V. Experiments in Urban Outdoor and Indoor Environments

The most important characteristic of a real 6DOF SLAM system using a stereo camera is the ability to solve the scale problem. Therefore we have focused on proving its applicability to recover the true scale in large environments. For this propose, we have used two 320x240 images sequences collected with a Point Grey Bumblebee stereo system at 25 fps (See Fig. 1). The system provides a 70 x 50 degree field of view per camera. This characteristic makes this work a real challenge, given that we have to ensure overlap between consecutive images to perform sequential matching. Also, the camera provides a baseline of 12cm, limiting the 3D point features initialization up to a distance non far from 2 meters. The sequences were processed with the proposed algorithms on a desktop computer with an Intel 4 processor at 2,4GHz.

The first sequence is composed of 3441 stereo pairs gathered in a public square of our home town. The full trajectory is approximately 140 meters long from the initial camera position. Figure 4 left shows the sequence of conditional independent local maps obtained with the technique provided in section III. Each map contains 100 features combining inverse-depth and 3D points. The total number of maps built during the stereo sequence is 11. The result of D&C without applying the loop closing constraint is shown in Fig. 4 middle. As it can be observed, the precision of the map obtained is good enough to almost align the first and last submaps after all the trajectory has been traversed, even without applying loop closing constraints. Fig. 4 right presents the final result after closing the loop.

Using Google maps tool we have checked that the map scale obtained and the trajectory followed by the camera is very close to the real scale. Fig. 6 illustrates comparative results. Moreover, it can be noticed that angles between the square sides and the shape of the walls of the surrounding environment have been finely captured.

The second experiment was taken inside one of our campus buildings in a walk of approximately 220 meters.

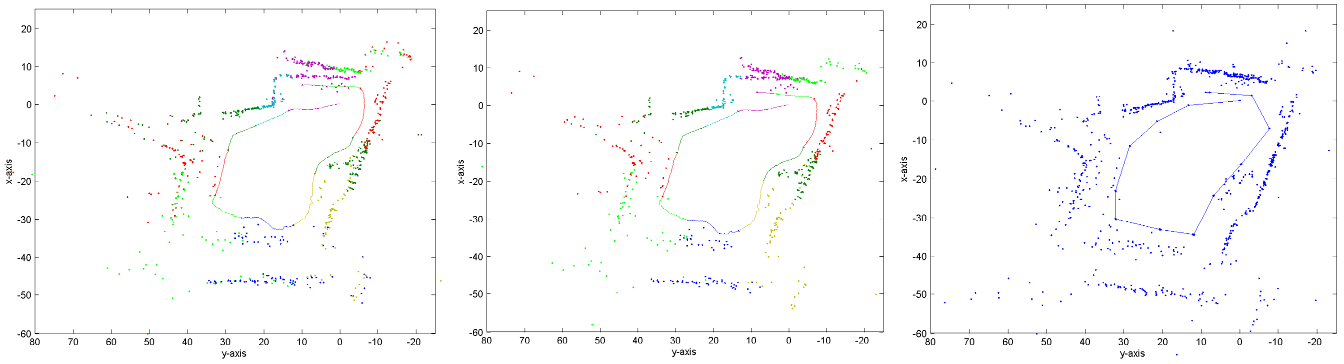

Fig. 4. 6DOF SLAM run in a public square. CI Local maps were carried to a common reference only to show the environment dimensions (left). D&C Updates were performed to correct the estimates (middle). Final result obtained when we impose loop closing constraints (right). Scale factor and camera positions are well recovered thanks to the combined observations of 3D points and inverse-depth points.

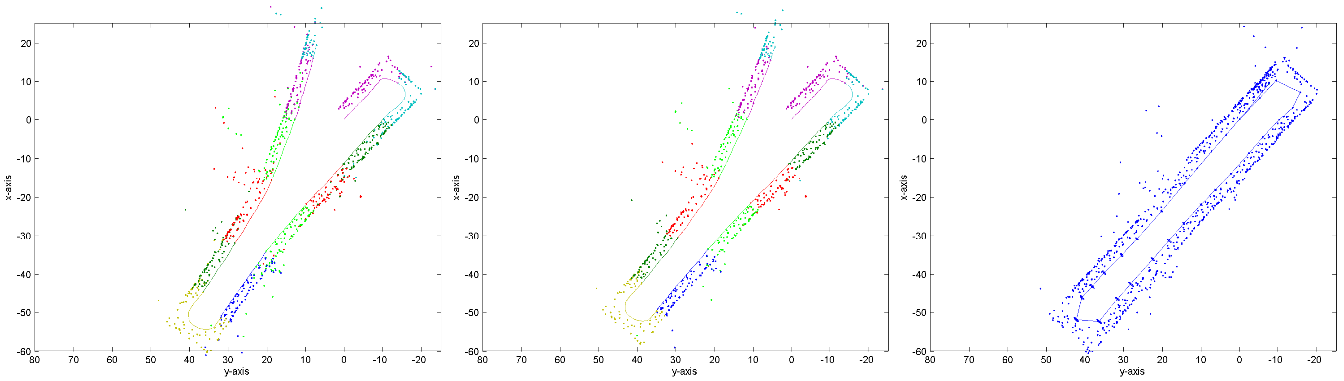

Fig. 5. 6DOF SLAM run in an indoor environment. Twelve CI local maps with 100 features each (left), D&C updates (middle) and final map estimate (right).

The same process was run in order to obtain a full map from 4081 stereo pairs. This environment has a particular degree of difficulty due to the presence of extend zones of glass windows such as offices, corridors and cafeterias. This can be noticed in the long distance points estimated in some of the maps, Fig.5 right. A bend appears when just inverse-depth points are extracted yielding to a deviation in the configuration of the final maps. Small corrections are observed when D&C is run (see Fig. 5 middle), but as shown in Fig.5 right, it was enough to make loop closing task possible with successful results.

We have also verified that our 6DOF SLAM system, even implemented in MATLAB, does not exceed 2 seconds per step, which is the worst case when building CI local maps. Fig. 7 shows how the running time system remains constant in most of the steps. Moreover, time peaks that appear when D&C takes place are below 6 seconds for the square experiment and 8 seconds for the indoor experiment, which are the maximum times required in the last step. These results point out that our system is suitable for a real time implementation.

VI. Conclusions

In this paper we have shown that the scale of large environments can be efficiently and accurately recovered using a stereo camera as the only sensor. During the experiments the camera was moved in 6DOF. One of the contributions of the paper is that information from features nearby and far from the cameras has been simultaneously incorporated to represent the 3D structure. Using close points provides scale information through the stereo baseline avoiding 'scale-drift', while inverse-depth points are useful to obtain angular information from distant scene points.

Another contribution of the paper is that combines two recent local maps techniques to improve consistency and reduce complexity. Using Conditionally Independent local maps, our system is able to properly share information related to the camera motion model, as velocities, and common features between consecutive maps. This points out that smoother transitions from map to map are achieved as well as better relative locations between local maps can be obtained. By means of the simplicity and efficiency of D&C, we can recover the full map allowing linear time execution. Thus, we can say that the combination of both techniques adds robustness to the estimation without sacrificing precision.

Although we are able to close the loop, the algorithm shown for loop closing strongly depends on extracting sets of features already stored in the map when the same area is revisited. It would be interesting to analyze other types of algorithms, for instance the image to map algorithm proposed

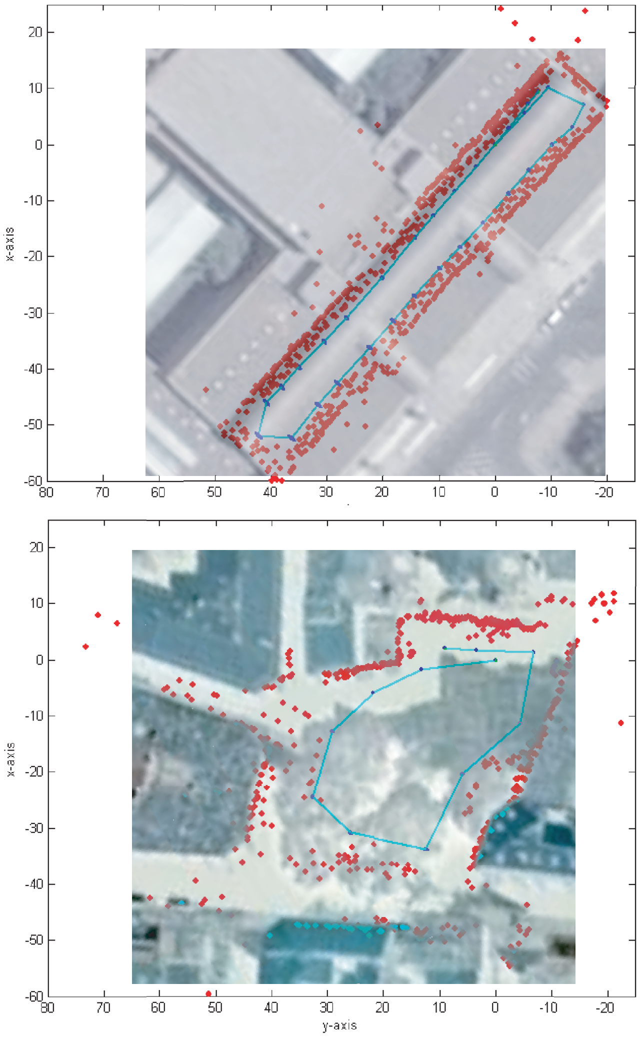

Fig. 6. Recovering true scale factor. The building environment (top) and The Public square (bottom) were reconstructed finely.

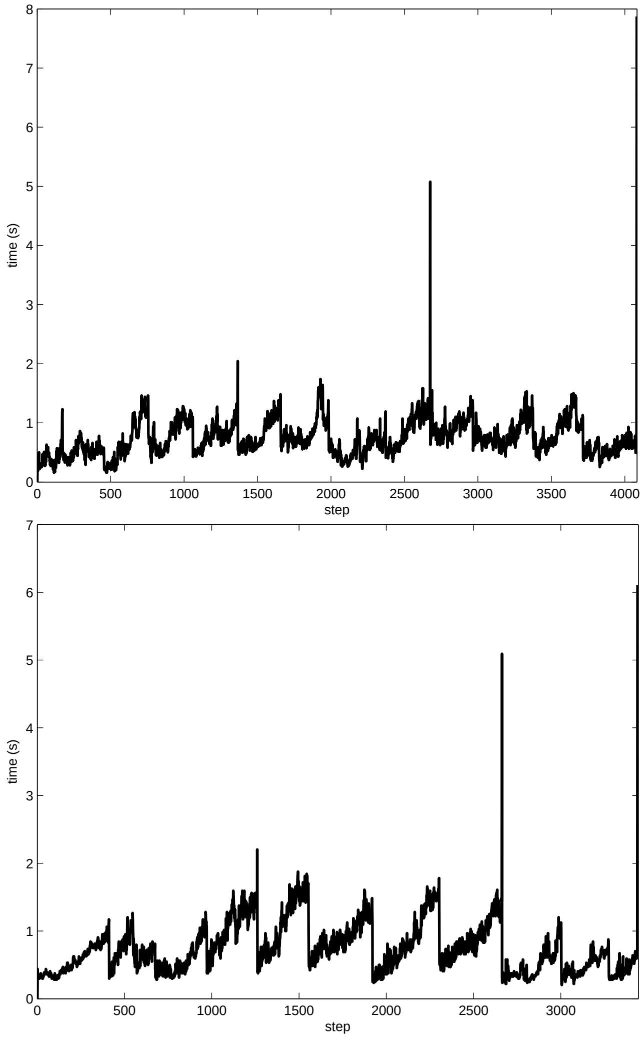

Fig. 7. Running time per step of all associated processes: Features extraction, Local Mapping, Data Association, D&C and Loop Closing. The building environment (top). The Public square (bottom).

in [12].

As future work, we will focus on compare our system with other stereo vision techniques as visual odometry. We are also interested in studying the fusion of the stereo camera with other sensors like GPS or inertial systems in order to compare the precision obtained. As well we have verified that the system is prone to the extraction of stable features. Therefore, we will analyze the effects produced when the features detector is changed.

References

[1] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, “Monoslam: Real-time single camera slam,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, June 2007.

[2] L. A. Clemente, A. J. Davison, I. D. Reid, J. Neira, and J. D. Tardós, “Mapping large loops with a single hand-held camera,” in *Robotics: Science and Systems, RSS*, June, 2007.

[3] L. Paz, P. Jensfelt, J. D. Tards, and J. Neira, “EKF SLAM Updates in O(n) with Divide and Conquer,” in *2007 IEEE Int. Conf. on Robotics and Automation*, Rome, Italy., April 2007.

[4] Z. Zhang and O.Faugueras, “Visual Odometry for Ground Vehicle Applications,” *International Journal of Computer Vision*, vol. 7, no. 3, pp. 211–241, 1992.

[5] D. Nistér, O.Naroditsky, and J. Bergen, “Visual Odometry for Ground Vehicle Applications,” *Journal of Field Robotics*, vol. 23, no. 1, 2006.

[6] A. Comport, E. Malis, and P. Rives, “Accurate quadri-focal tracking for robust 3d visual odometry,” in *IEEE Int. Conf. on Robotics and Automation, ICRA*, April, 2007.

[7] P. Pinies and J. D. Tardós, “Scalable slam building conditionally independent local maps,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*, November, 2007.

[8] A. Davison, “Real-time simultaneous localisation and mapping with a single camera,” Oct. 2003.

[9] J. M. M. Montiel, J. Civera, and A. J. Davison, “Unified inverse depth parametrization for monocular slam,” *Proceedings of Robotics: Science and Systems*, 2006 August.

[10] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[11] L. M. Paz, J. Guivant, J. D. Tardós, and J. Neira, “Data association in linear time for divide and conquer slam,” in *Robotics: Science and Systems, RSS*, June, 2007.

[12] B. Williams, P. Smith, and I. Reid, “Automatic relocalisation for a single-camera simultaneous localisation and mapping system,” Roma, Italy, April 2007, pp. 2784–2790.