Entropy Minimization SLAM Using Stereo Vision*

Juan Manuel Sáez and Francisco Escolano
Robot Vision Group
Departamento de Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante, Ap.99, E-03080, Alicante, Spain
{*jmsaez, sco*}*@dccia.ua.es*

***Abstract*— In this paper we present an information-based approach to solve the SLAM problem using stereo vision. This approach results for an improvement, in terms of both efficiency and robustness, of our early multi-view ICP randomized algorithm. Instead of minimizing an ICP-based cost, we propose the minimization of the entropy of the 2D distribution induced by the projection of the 3D point cloud. In addition we embed both the egomotion/action estimation algorithm which precedes global rectification and the new global rectification algorithm in an autonomous exploration schema. We assume plane-parallel environments and, for the sake of efficiency, we also assume a flat floor and a fixed stereo camera mounted on the robot. We show successful experiments both under tele-operating the robot and under autonomous navigation.**

Index Terms*— *SLAM, Stereo Vision, Navigation, 3D Mapping, Entropy Minimization.

I. Introduction

Solving the SLAM problem is key for endowing robots with real autonomous capabilities. In this regard, when the used sensors yield dense enough data it is reasonable to use mobile robots to build 3D maps of the environment. However, such a task must be done both robustly and efficiently and, thus, the evaluation of both the convergence of the map and the computational requirements of the algorithm is necessary [1]. This reasoning motivates the SLAM approach introduced in this paper which is a significant improvement, in terms of convergence and computational requirements, of an early approach [2] which is focused on the use of stereo cameras as unique 3D sensors for localization and mapping.

Several stereo-based mapping and navigation approaches have been proposed in the past. In [3], where 3D information yields 2D maps, these maps are represented with a occupancy grid models instead of using the unpractical 3D occupancy grid model [4]. In other cases, like in [5], stereo data even allows the computation of 3D planes although some manual guidance is necessary when there is no data. In [6], stereo vision is fused with inertial information in order to recover 3D segments. On the other hand, in [7][8] 3D landmarks based on scale-invariant image features are used to compute the map. Such a computation relies on estimating the ego-motion of the robot, tracking the landmarks using the odometry for prediction, and finally superposing the landmarks to obtain the map. Such an approach is not globally consistent because it only relies on local estimations. This is why it is extended in [9], although the global consistency proposed exploits the closing-the-loop constraint for performing backward corrections.

On the other hand, existing approaches using laser range finders sensors, have evolved towards obtaining more and more compact 3D models made up of planar surfaces [10][11][12][13][14]. In such approaches, the problem of simultaneously computing the map and robot poses is formulated in terms of maximizing a log-likelihood function and then using an EM-like algorithm to obtain, at least, a local maximum. In other approaches, range finders are used in combination with cameras, following the idea of complementing sensor capabilities. For instance, in [15] a range finder is combined with eight CCD cameras to compute a mesh model of the environment from a large number of overlapped 3D images. In [16][17], a range scanner is combined with a single camera through fusing the results of 2D and 3D matching and planar segmentation processes, and a high-level planning system obtains the best next view for acquiring the model of the environment.

Our former approach to SLAM with stereo vision relies on an egomotion/action estimation algorithm which performs a local 3D alignment of a reduced cloud of points using the local appearance of their projections in the reference images. Given a trajectory, that is, a sequence of robot actions and observations (reduced 3D clouds), a multi-view ICP algorithm obtains the global rectification of the map (consistent registration as in [18]) through the minimization of an energy function. The first term of that energy function is an ICP-like term relying on distances between the points of a given view to the closest points in the rest of the map, and the second term enforces the assumption that the environment is plane-parallel, that is, all planes are either parallel or orthogonal. For the sake of efficiency, which is reasonable when working in indoor environments, we also assume that the robot is confined to the XZ plane. We make no additional assumptions regarding cyclic robot trajectories.

Although the randomized multi-view ICP algorithm produces acceptable results its robustness and efficiency may be improved. First, evaluating the first term of our former energy function has a quadratic complexity with the number of points and this forced us to use a reduced

*This research is partially funded by the project TIC2002-02792 of the Spanish Government

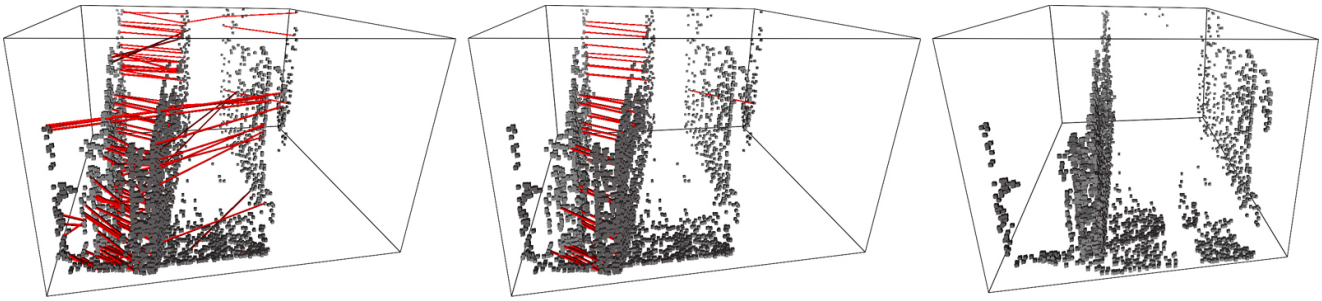

Fig. 1. Egomotion/action estimation. Left: Initial alignment has 99 matches. Center: After the matching refiniment only 38 matches are retained. Right: From this latter matches we compute the optimal transformation which is $\delta x_t = -0.071m, \delta z_t = 0.564m, \delta\theta_t = 11.960^o$.

point cloud. Second, when minimizing this term we have detected several local minima that induce alignment fails in some situations. Instead of using term inspired by the ICP algorithm [19] or by its variations [14][20], here we propose a simple information-based criterion that may be evaluated in a linear time and avoids the latter local minima. In addition, we embed both the egomotion and the new global rectification algorithm in an autonomous exploration algorithm.

The technical details of the approach are described in Section II(egomotion) and in Section III (map building and rectification). In Section IV we embed the mapping algorithm in an autonomous exploration strategy. In Section V we present some indoor mapping experiments performed both by tele-operating a mobile robot, equipped with a stereo camera, that implements our new rectification algorithm and by allowing such a robot to perform autonomous exploration while solving the SLAM problem. Finally, in Section VI we present our conclusions and future work.

II. Egomotion/Action Estimation

The goal of egomotion/action estimation is to estimate the incremental action $\mathbf{a}_t = [\delta x_t, \delta z_t, \delta\theta_t]^T$ describing how the current pose of the robot $\mathbf{p}_t = [x_t, z_t, \theta_t]^T$ the $t-$th (where the two first components are the coordinates in the ZX horizontal-frontal plane, and the third one is the orientation with respect to the vertical axis Y) has been derived from the previous one $\mathbf{p}_{t-1}$, but only using visual cues. Here, and for the sake of completeness, we summarize the four stages of our algorithm (see Fig. 1) with minor modifications with respect to our early work (see [2] for more details):

A. Feature Extraction

Given the 3D point cloud $\mathbf{C}_t$ observed from the $t-$th pose, and for the sake of both efficiency and robustness, we retain only those points $\mathbf{M}_t = [x_t, y_t, z_t]^T \in \mathbf{C}_t$ (where OZ is the focal axis and XY is the image plane) whose projections in the $t-$th right stereo image (reference image) are associated to strict local maxima of the image gradient. Furthermore, and in order to consider the acquisition error, we discard 3D points too far from the camera. The retained points define the constrained cloud $\tilde{\mathbf{C}}_t$ that will be used in this algorithm.

B. Feature Matching

Under the assumptions of flat floor and fixed camera on the robot, a point $\mathbf{M}_t$ should match another point $\mathbf{M}_{t-1}$ lying in the same Y plane, with a given tolerance δ. This reduces considerably the complexity of our matching process. We maximize the score

$$S(\mathbf{M}_t, \mathbf{M}_{t-1}) = |\rho(\mathbf{Z}_t, \mathbf{Z}_{t-1})|, \tag{1}$$

being $\rho(\mathbf{Z}_t, \mathbf{Z}_{t-1}) \in [-1, 1]$ the Pearson coefficient (illumination invariance) of the random variables associated to the grey intensities of the log-polar mappings (local orientation invariance) of the windows with size $|\mathbf{W}|$ centered on the projections of the points in their respective reference images. Furthermore, we filter matching candidates with low strength, low distinctiveness with respect to the second best candidate, or non bidirectional.

C. Matching Refinement

Despite considering the three later conditions, the matching process is prone to outliers which must be identified and removed. For instance, if the $i-$th point $\mathbf{M}_t^i$ of $\tilde{\mathbf{C}}_t$ matches the $j-$th point $\mathbf{M}_{t-1}^j$ of $\tilde{\mathbf{C}}_{t-1}$, and similarly $\mathbf{M}_t^k$ matches $\mathbf{M}_{t-1}^l$, we will remove the first match if the following quantity is high enough:

$$D_{ij} = \frac{\sum_k \sum_l D_{ikjl}}{|\mathcal{M}|}, \tag{2}$$

where $\mathcal{M}$ is the current set of matches, and D_{ikjl} is the maximum ratio between $||\mathbf{M}_t^i$ - $\mathbf{M}_t^k||$, and $||\mathbf{M}_{t-1}^j$ - $\mathbf{M}_{t-1}^l||$, which must be close to the unit in good matches (similar relative distances). Then, in the *leaving-the-worst-out* process, matches with higher D_{ij} are iteratively considered for removing. As the values of D_{ij} tend to be uniform as we remove outliers, we stop the process when either such a deviation reaches σ_{min}, being σ_{min} sufficiently small, or a minimum number of matches $|\mathcal{M}|_{\min}$ is reached. Comparing this criterion with the statistical filter proposed in [21], we consider structural differences within the views (independently of the relative position between

them) instead of considering structural differences between the views.

D. Action/Transformation Estimation

Once we perform the latter matching refinement, which in practice is robust enough to avoid a interleaved EM-like process, we proceed to estimate, the rotation $\mathbf{R}_t$ and translation $\mathbf{t}_t$ associated to action $\mathbf{a}_t$. Given that each point $\mathbf{M}_t^i \in \tilde{\mathbf{C}}_t$ matches $\mathbf{M}_{t-1}^j \in \tilde{\mathbf{C}}_{t-1}$ the optimal action is the one minimizing the usual quadratic energy function

$$E(\mathbf{R}_t, \mathbf{t}_t) = \sum_i \sum_j \mathbf{B}_{ij} ||\mathbf{M}_{t-1}^j - (\mathbf{R}_t \mathbf{M}_t^i + \mathbf{t}_t)||^2, \quad (3)$$

being $\mathbf{B}_{ij}$ binary matching variables (1 when $\mathbf{M}_t^i$ matches $\mathbf{M}_{t-1}^j$ and 0 otherwise). In our early version of the algorithm we performed such a minimization through conjugate gradient descent, but in the present version we use a faster randomized method which selects pairs of matched points to estimate a given transformation and then such a transformation is used to align both clouds. The transformation with better alignment is retained. This procedure could be seen as a simplified version of the RANSAC algorithm [22].

III. Map Building and Rectification

Given a robot trajectory $\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_{N-1}$ of size N, and a sequence of estimated actions $\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_{N-1}$ with length N, being $\mathbf{a}_0 = \mathbf{p}_0$, an initial approximation of the 3D map comes from superposing all the point clouds with respect to the referential pose $\mathbf{p}_0$. We call the aggregation of all observations in a common reference system (initial map) as $\mathbf{A}$ when it relies on the complete point clouds. However, as this map accumulates the errors produced at local action estimations (errors due to the latter algorithm, or even to the absence of 3D cues when non-textured parts of the environment are observed), it is desirable to provide a consistency criterion and an updating strategy that exploits it in order to obtain a globally-consistent map.

In our former approach, we formulated the consistency criterion in terms of the sum of the lack of consistency of all views, and such a lack of consistency was measured as the averaged distance between its points and their closest neighbors lying in the same Y plane (the robot is confined to the XZ plane) and belonging to the rest of views defining the constrained map. However, as the latter criterion relies on distances to the closest points it has two drawbacks: First, the complexity of computing such distances is quadratic with the number of points in the constrained map. This complexity forced us to use the constrained point cloud $\tilde{\mathbf{A}}$ instead of the complete one. Moreover, such a number increases significantly with a new observation slowing down the rectification process. Second, we found that the minimization of such criterion in a multiple-view scheme may fall into local minima (see Fig. 2, center) and, thus, it may fail to align walls not closer enough. In order to deal both with the quadratic complexity and the local minima we propose here a simple criterion relying on measuring the information contained in the projection of the huge 3D point cloud $\mathbf{A}$ on the XZ plane. The redefinition of such criterion has important implications in the optimization procedure.

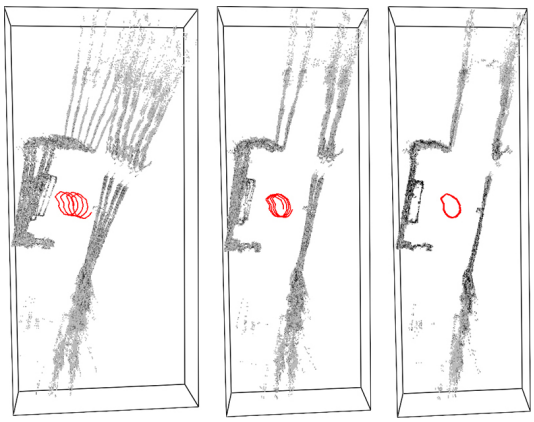

Fig. 2. Comparison between the closest-neighbors measure (multiple-view ICP) and the 2D entropy criterion: Original map (left), multiple-view ICP (center), and entropy-based criterion (right).

A. Consistency Criterion

Our information maximization criterion for measuring the global consistency of the complete (not reduced) point cloud of the current map $\mathbf{A}$ is the minimization of the energy:

$$E(\mathbf{A}) = H(\mathbf{q}_{XZ}) + \mu(H(\mathbf{q}_X) + H(\mathbf{q}_Z)), \quad (4)$$

where $H(.)$ denotes the entropy of the argument. In the latter criterion we consider three probability distributions. First (and following in this case the assumption that the robot is confined to the XZ plane) $\mathbf{q}_{XZ}$ is the probability distribution $\mathbf{q}_{XZ}(x,z) = q(x,z)/\sum_x \sum_z q(x,z)$ associated to normalizing, after a proper discretization of the XZ plane, and for each pair $x \in X$ and $z \in Z$, the sum $q(x,y)$ of all points $\mathbf{M}^i = [x^i, y^i, z^i]$ in the map $\mathbf{A}$ satisfying both $x^i \approx x$ and $z^i \approx z$. The role of this first term is to avoid the local minima of the closest-neighbor criterion when multi-view ICP minimization is applied, and also to yield a linear computational complexity, when evaluating the criterion, instead of a quadratic one (with respect to the number of points in the map). For instance, in Fig. 3, we compare the entropy of a bad alignment with that of a good one. In practice, and prior to computing $\mathbf{q}_{XZ}$, the sum $q(x,z)$ may be smoothed by a convolution with a Gaussian kernel of variance σ_{XZ} in order to combine the projections of walls that are close enough with respect to that variance.

On the other hand, the second term of Eq. 4, $\mu(H(\mathbf{q}_X) + H(\mathbf{q}_Z))$ where $\mu \geq 0$ is a scaling factor, is only applicable to plane-parallel environments, that is, in environments where the main planes (walls, doors, floor, ceiling, and

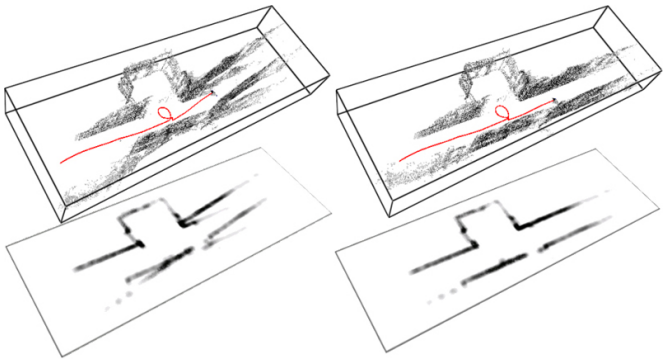

Fig. 3. Example of a map after 43 observations. Left: the entropy of a bad alignment (see the 2D distribution below the 3D map) is 6.891403. Right: the entropy of a good alignment is 6.730805. The entropy range depends on the map.

so on) are either parallel or orthogonal. This allows, for instance, to correct a typical straight corridor that appears slightly curved if we only minimize the sum of local energies. In order to do so, we compute the marginal distributions $\mathbf{q}_X(x) = \sum_z \mathbf{q}_{XZ}(x,z)$ and $\mathbf{q}_Z(z) = \sum_x \mathbf{q}_{XZ}(x,z)$. Intuitively, if a corridor is perfectly orthogonal to the X axis, we will observate two well-defined peaks in the projection and $H(\mathbf{q}_X)$ will be minimal with respect to any other rotation (see Fig. 4). Furthermore, the sum $H(\mathbf{q}_X) + H(\mathbf{q}_Z)$ is maximal when either the corridor is orthogonal to X or to Z. Therefore, this second term, that we call alignmnent term prefers maps yielding peaked projections and this is why curved corridors may be rectified. In order to accelerate the process, the first observation (reference system) should be approximately aligned with the main building directions.

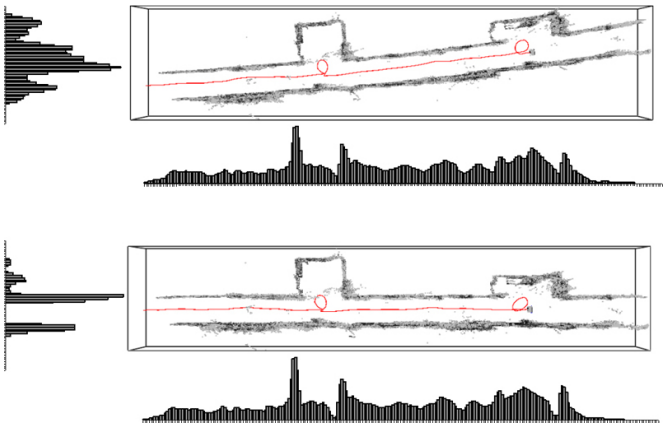

Fig. 4. Alignment term.

B. Quasi-random Update

The underlying idea of our map-updating strategy is to modify all actions $\mathbf{a}_t$ simultaneously (including $\mathbf{a}_0$ because $\mathbf{p}_0$ could share information with $\mathbf{p}_{N-1}$, for instance, when the trajectory is cyclic) in order to obtain a new map $\mathbf{A}^{new}$. However, in practice, and in order to ensure the convergence of the optimization problem, we only modify simultaneously $K < N$ actions. Such actions are selected randomly, and once a given action $\mathbf{a}_t$ is selected, the new action $\mathbf{a}_t^{new}$ is equal to the given one on average but there is an error defined by ϵ_t. That is, $\mathbf{a}_t^{new}$ is a random variable following the Gaussian distribution $N(\mathbf{a}_t, \epsilon_t)$ with $\epsilon_t = p_t \mathbf{S}_t$, where $\mathbf{S}_t$ is diagonal 3×3 matrix defined by the variances $\sigma_{\delta x}$, $\sigma_{\delta z}$ and $\sigma_{\delta\theta}$, whose scale must be carefully specified. Furthermore, these variances are modulated by p_t, a probability resulting from applying the *past usefulness rule*, that is, actions which yielded an improvement of the information (entropy reduction) in the past will have a higher amplitude (see the algorithm in Fig. 5 where we implement this idea through a voting scheme). Such modulation of the amplitude allows to search in the neighborhood of useful views in the past.

GLOBAL RECTIFICATION ALGORITHM
Input: Map $\mathbf{A}$, Actions $\mathbf{a} = \{\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_{N-1}\}$
Output: Optimal actions $\mathbf{a}^* = \{\mathbf{a}_0^*, \mathbf{a}_1^*, \ldots, \mathbf{a}_{N-1}^*\}$
// Initialize votes and optimal actions
$v_t \leftarrow 1, t = 0, 1, \ldots, N-1$;
$\mathbf{a}_t^* \leftarrow \mathbf{a}_t, t = 0, 1, \ldots, N-1$;
// # Global iterations = # Iterations without changes = 0
$I_g \leftarrow I_{nc} \leftarrow 0$;
// Initialize optimal energy with the initial map
$E^* \leftarrow E(ComputeMap(\mathbf{A}, \mathbf{a}))$;
// Main loop
while $I_g < G_{max}$ and $I_{nc} < NC_{max}$ **do**
  // Select K views randomly
  // First set to zero the indicator variables
  $s_t \leftarrow 0, t = 0, 1, \ldots, N-1$;
  // Then select an indicator to set
  while $\sum_{t=0}^{N-1} s_t < K$ **do**
    $s_{random[0\ldots N-1]} \leftarrow 1$;
  endwhile
  // Update probabilities attending to votes
  $p_t \leftarrow s_t \frac{v_t}{\sum_{r=0}^{N-1} v_r s_r}, t = 0, 1, \ldots, N-1$;
  // Propose modifications for selected actions
  $\mathbf{a}_t^{new} \leftarrow Sample(N(\mathbf{a}_t, \epsilon_t)), t = 0, 1, \ldots, N-1$;
  // Evaluate new energy and map
  $E^{new} \leftarrow E(ComputeMap(\mathbf{A}, \mathbf{a}^{new}))$;
  // Update energy and votes if proceeds
  if $E^{new} < E^*$ **then**
    $E^* \leftarrow E^{new}$;
    $I_{nc} \leftarrow 0$;
    // Past usefulness rule
    $v_t \leftarrow v_t + s_t, t = 0, 1, \ldots, N-1$;
    // Update optimal actions
    $\mathbf{a}_t^* \leftarrow \mathbf{a}_t^{new}, t = 0, 1, \ldots, N-1$;
  else
    $I_{nc} \leftarrow I_{nc} + 1$;
  endif
  $I_g \leftarrow I_g + 1$;
endwhile
end

Fig. 5. Global rectification algorithm

Considering all new actions $\mathbf{a}_0^{new}, \mathbf{a}_1^{new}, \ldots, \mathbf{a}_{N-1}^{new}$ simultaneously in the algorithm implies considering new poses $\mathbf{p}_0^{new}, \mathbf{p}_1^{new}, \ldots, \mathbf{p}_{N-1}^{new}$, and consequently a new map $\tilde{\mathbf{A}}^{new}$. Such a map is accepted if it yields the energy, that is, if $E(\mathbf{A}^{new}) < E(\mathbf{A})$. Otherwise, none of the new actions are applied, we retain $\mathbf{A}$, and a new iteration of the algorithm begins. We finalize when the stabilization of the global energy is detected (a given number of iterations without changes is reached) of a maximum number of iterations is reached.

IV. Autonomous Exploration

Solving the SLAM problem in while the robot is exploring the environment requires integrating both the egomotion and global rectification algorithms with path-planning facilities attending to the current state of the robot and its knowledge of the environment. In this paper we adapt to this problem the computation of the classical distance transform (see [23] and potential fields [24], and also [25] for early experiments with stereo data). Given a 2D evidence grid resulting from projecting and discretizing the current 3D map, we threshold the grid, compute the Minkowsky's addition and, assuming that the robot's goal is to escape from the map (if possible), we proceed to label each cell $\mathbf{c}_i$ as GOAL, if it lies in the map's bounding box, OBSTACLE, and FREE. Then (see Fig. 6) we from any free cell $\mathbf{c}_i, l(\mathbf{c}_j) =$ FREE its distance to the closest obstacle and we obtain, through a propagation process, the obstacle distance transform: $\mathbf{D}^{obst}(\mathbf{c}_i) = ||\mathbf{c}_i - \mathbf{c}^{closest}||$ where $\mathbf{c}^{closest}$ minimizes $||\mathbf{c}_i - \mathbf{c}_j||$ among those cells with $l(\mathbf{c}_j) =$ OBSTACLE. Similarly, we obtain the goals distance transform $\mathbf{D}^{goal}(\mathbf{c}_i)$. Having the latter distance transforms the potential field landscape is defined by

$$\mathbf{U}(\mathbf{c}_i) = \mathbf{D}^{goal}(\mathbf{c}_i) - \lambda \mathbf{D}^{obstacle}(\mathbf{c}_i), \qquad (5)$$

where $\lambda \geq 0$ is a weighting factor designed so that the robot satisfies the goal while travelling as far as possible from the obstacles. The path to the closest goal is then obtained through gradient descent.

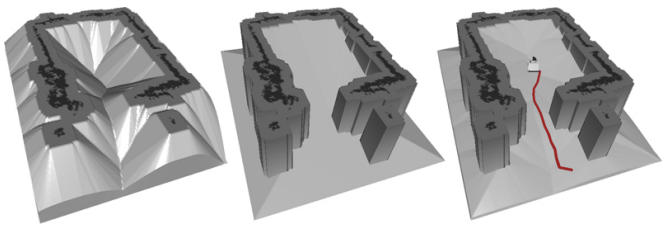

Fig. 6. Distance transforms and potential field landscape. Left: $-\mathbf{D}^{obstacle}$ (inverted for visualization purposes). Center: $\mathbf{D}^{goal}$. Right: $\mathbf{U} = \mathbf{D}^{goal} - 0.5 \times \mathbf{D}^{obstacle}$ and gradient descent.

Considering the latter path-planning approach and integrating it with the egomotion and rectification algorithms described above, the resulting autonomous exploration algorithm for SLAM (see Fig. 7) begins by performing a circular trajectory for getting an initial map. Then, such map is rectified and the call $ComputeNewPath(\mathbf{A}, \mathbf{a})$ returns a proposed path for exploration which initially feeds the main loop of the algorithm which consists of: (i) extract, and apply, a new action for the proposed path; (ii) compute a new action through egomotion; (iii) test the reliability of egomotion and discard the new action if proceeds; (iv) test whether global rectification proceeds (when a given number of observations are performed or egomotion fails); (v) Test, , whether the last action was the last in the proposed path or whether that path is not save as indicates $FutureCollision(\mathbf{A}, \mathbf{a}^{prop})$. If so, compute a new path. The call $ComputeNewPath(\mathbf{A}, \mathbf{a})$ will return a void path when the robot reaches a plateau in the potential field landscape, that is, when the map is closed.

AUTONOMOUS EXPLORATION ALGORITHM
Input: None
Output: Full map $\mathbf{A}$
// First observation and action
$\mathbf{C}_0 \leftarrow CaptureStereoView()$;
$\mathbf{a}_0 \leftarrow (0, 0, 0)$;
$\mathbf{a} \leftarrow \mathbf{a}_0$;
// R observations to compute the map around the robot
for $t = 1$ **to** R
  $TurnRobot(360/R)$;
  $\mathbf{C}_t \leftarrow CaptureStereoView()$;
  $\mathbf{a}_t \leftarrow EgoMotion(\tilde{\mathbf{C}}_t, \tilde{\mathbf{C}}_{t-1})$;
  $\mathbf{a} \leftarrow Add(\mathbf{a}, \mathbf{a}_t)$;
endfor
// Rectification and new path computation
$\mathbf{a} \leftarrow GlobalRectification(\mathbf{A}, \mathbf{a})$;
$\mathbf{a}^{prop} \leftarrow ComputeNewPath(\mathbf{A}, \mathbf{a})$;
// Main loop
while $\mathbf{a}^{prop} \neq \phi$ **do**
  // First action of the new path
  $\mathbf{a}_{new} \leftarrow ExtractFirst(\mathbf{a}^{prop})$;
  $MoveRobot(\mathbf{a}_{new})$;
  // Observation and action of the new pose
  $\mathbf{C}_t \leftarrow CaptureStereoView()$;
  $\mathbf{a}_t \leftarrow EgoMotion(\tilde{\mathbf{C}}_t, \tilde{\mathbf{C}}_{t-1})$;
  // If ego-motion fails, take the proposed action
  if $\neg Reliable(\mathbf{a}_t)$ **then**
    $\mathbf{a}_t = \mathbf{a}_{new}$;
  endif
  // Global rectification each S_{rec} actions
  if $t\%S_{rec} = 0$ **or** $\neg Reliable(\mathbf{a}_t)$ **then**
    $\mathbf{a} \leftarrow GlobalRectification(A, a)$;
  endif
  // If the path is void or produces a collision,
  // compute a new one
  if $\mathbf{a}^{prop} = \phi$ **or** $FutureCollision(\mathbf{A}, \mathbf{a}^{prop})$ **then**
    $\mathbf{a}^{prop} \leftarrow ComputeNewPath(\mathbf{A}, \mathbf{a})$;
  endif
  $t \leftarrow t + 1$
endwhile

Fig. 7. Autonomous Exploration algorithm

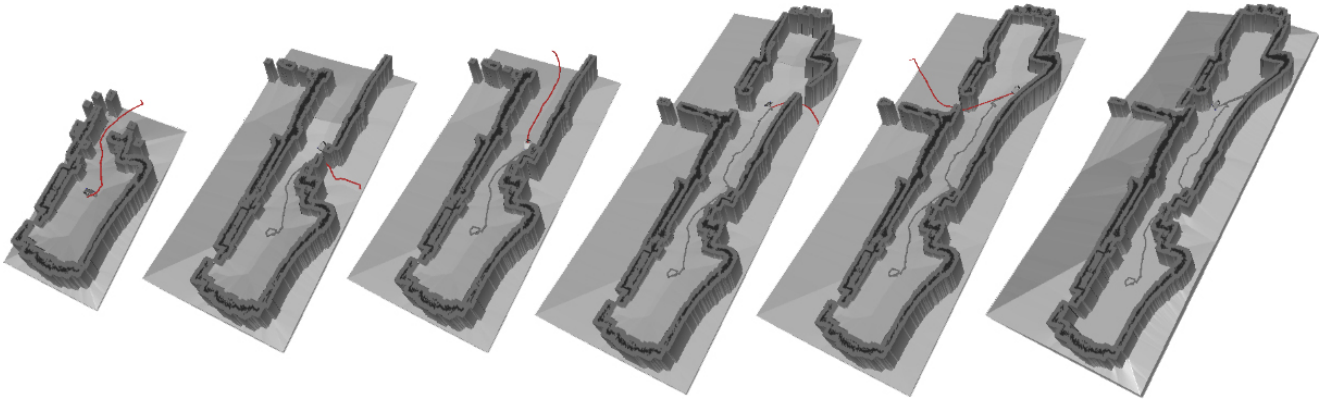

Fig. 8. Autonomous exploration SLAM. From left to right: several partial paths and their respective maps. The final map is built in 144 observations. We show both the past path and the proposed one. We ensure that collision is avoided.

Fig. 9. Our mobile robot equipped with the stereo system.

In Fig. 8 we show a complete experiment of SLAM under autonomous exploration. First, the robot takes a circular path to obtain the initial map. Then, it computes the optimal path which is applied until a potential collision is discovered. As part of the environment is observable through windows, but it is not reachable, the map is closed without exploring the corridor in the middle of the map.

V. Experimental Results

For our mapping experiments we use a ER1 robot equipped with a trinocular Digiclops stereo system, a laptop with a Pentium IV/2.4GHz processor, and a wireless ethernet system for tele-operation when necessary (see Fig. 9). We obtain 320×240 stereo images producing clouds of 10.000 points on average which are typically reduced to 500 points in the constrained clouds. Given our previous experimental evaluations of the 3D estimation error for the Digiclops system, our maximum range is 8 meters, being the averaged error associated to such a distance below 0.75 meters.

The parameters for action estimation are: $|\mathbf{W}| = 7 \times 7$ (the size of the appearance windows), $\delta = 0.05$ meters (range for considering that two points have similar height), $S_{min} = 0.8$ (minimal strength of a match), $R_{min} = 0.95$ (minimal distinctiveness ratio), $\sigma_{min} = 0.005$ and $|\mathcal{M}|_{\min} = 10$ (variance limit and minimal number of matches for stopping the leaving-the-worst-out process).

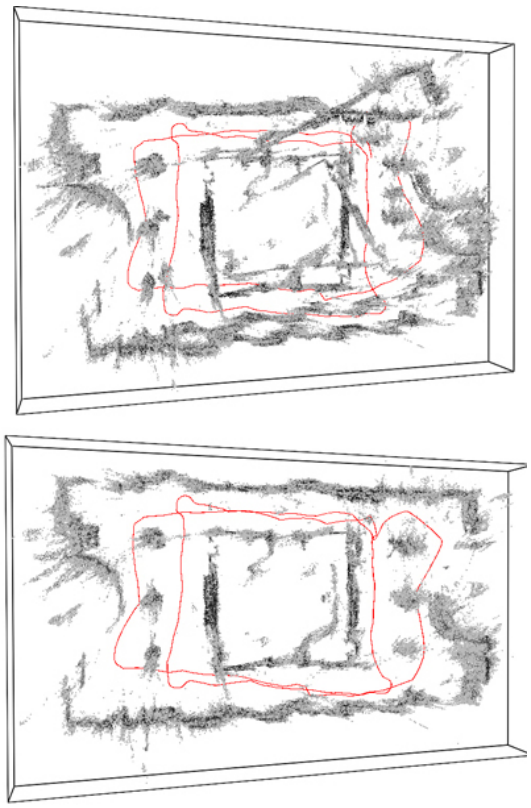

Fig. 10. Economics Hall. Double loop of 308 observations after travelling 134.57m. Egomotion (top) and global rectification each 10 observations (bottom). The repeated information allows a good alignment.

The parameters for map building and rectification are: $\sigma_{\delta x} = \sigma_{\delta z} = 0.016$ meters, $\sigma_{\delta\theta} = 2.86$ degrees. The fraction K/N of simultaneously selected actions in the quasi-random update process is between 0.1 and 0.15 (then for typically registrations with N from $100-300$ views, K is in

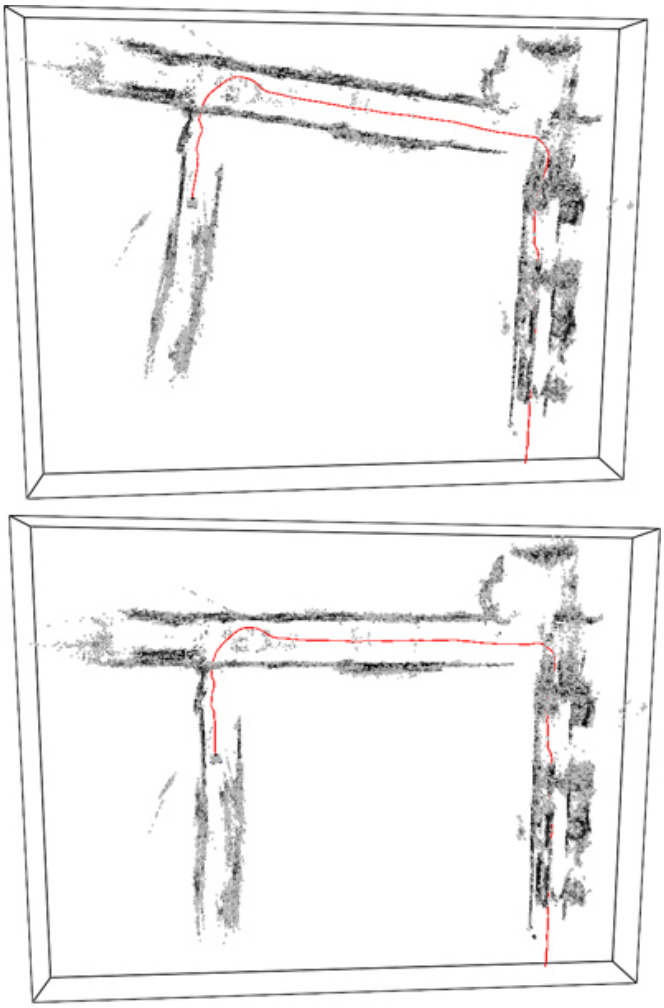

Fig. 11. Corridor. Trajectory of 148 observations after travelling 35.06m. This is a plane-parallel environment and the good alignment obtained relies on this assumption.

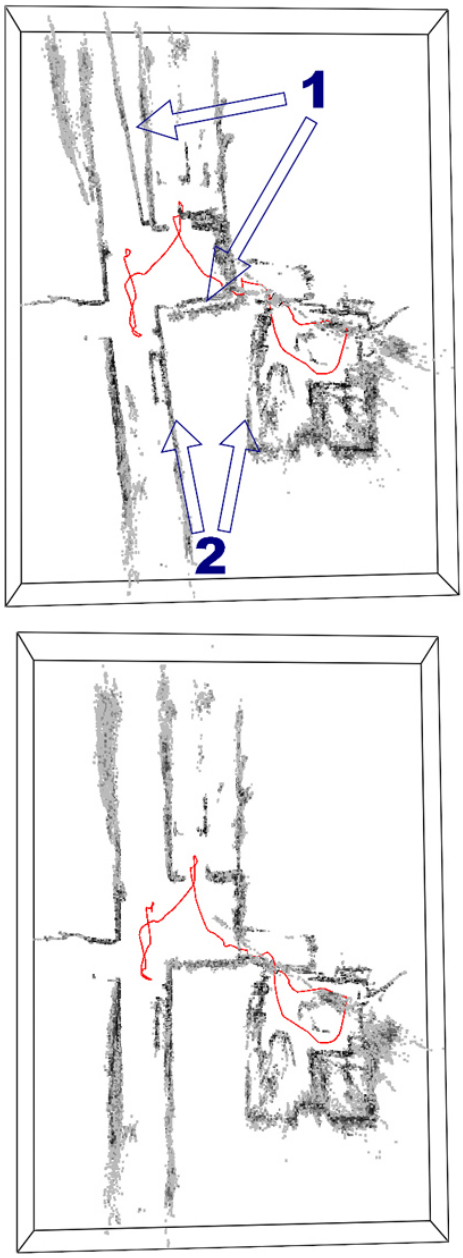

Fig. 12. Corridor and room. Trajectory of 151 observations along 26.32m. Repeated walls (1) are corrected, and also the room (2) is seen to be parallel to the corridor.

the range $15-45$). Regarding the plane-parallel enforcing term, the constant μ for weighting the sum of entropies is set to 0.5. Both the 2D map for computing entropies and the potential field landscape have a resolution of 5cm. In the autonomous navigation algorithm, the number of views in the initial circular path is $R = 20$ (rotations of 18^o), and the number of observations between performing a global rectification is $S_{rec} = 10$. When computing the obstacles, we assume that the number of occurrences of a given 3D point in the 2D map must be greater than 50.

Our code is written in C++ and we have used the Intel Performance Primitives for accelerating computations. The averaged time for action estimation was reduced from 323 ms in the early version to 110ms. The averaged time for evaluating the entropy ranges from 39ms to 404ms depending on the size of the map. The global rectification algorithm takes from 3 secs, for a map of 20 views, to 25 secs for maps of 200 views, or even to 35 secs for maps of 300 views with very bad initial alignments.

We have performed two types of experiments: teleoperation experiments (see Figs. 10,11,12) and exploration experiments (see Fig. 13, the same environment as in Fig. 8). In all experiments we show the result of using only the egomotion algorithm with the result of using the global rectification. The details of each experiment are described in the corresponding captions.

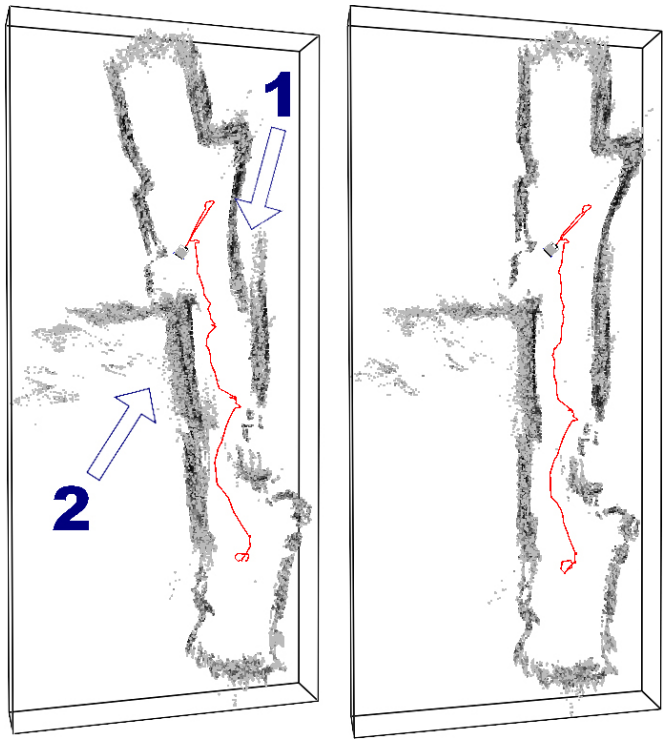

Fig. 13. Autonomous exploration. Trajectory of 144 views along 24.82m. A curved wall is aligned (1). There are also many cases in which the egomotion fails due to the existence of windows (2), and these windows let the robot see the exterior wall of a unreachable corridor.

VI. Conclusions and Future Work

We have proposed an information-based SLAM approach to find globally consistent 3D maps using stereo vision, and we have successfully tested it in many environments, both with tele-operation and with autonomous navigation. However, there many aspects to solve, and our current and future research includes: (i) extending the approach for more unconstrained egomotion and environment conditions, (ii) introduce tolerance to dynamic entities in the environment (like people walking), and (iii) compute compact 3D maps with primitives like planes and cylinders with active stereo.

References

[1] A. Martinelly, N. Tomatis, and R. Siegwart. *Open Challenges in SLAM: An Optimal Solution Based on Shift and Rotation Invariants.* Proceedings of ICRA'04: IEEE International Conference on Robotics and Automation (2004).

[2] J.M. Sáez and F. Escolano. *A Global 3D Map-Building Approach Using Stereo Vision*. Proceedings of ICRA'04: IEEE International Conference on Robotics and Automation (2004).

[3] D. Murray and J. Little. *Using real-time stereo vision for mobile robot navigation*. Autonomous Robots, Vol. 8, N. 2 (2000) 161-171.

[4] H. Moravec. *Robot spatial perception by stereoscopic vision and 3D grids*. Technical Report, The Robotic Institute, CMU (1996).

[5] M.B.L. Iocchi and K. Konolige. *Visually realistic mapping of a planar environment with stereo*. In Proceedings of ISER'00: International Conference on Experimental Robotics (2000).

[6] J. Lobo, C. Queiroz, and J. Dias. *World feature detection using stereo vision and inertial sensors*. Robotics and Autonomous Systems 44 (2003) 69-81.

[7] S. Se, D. Lowe, J. Little. *Vision-based mobile robot localization and mapping using scale-invariant features*. In Proceedings of ICRA'01: IEEE International Conference on Robotics and Automation (2001).

[8] S. Se, D. Lowe, J. Little. *Mobile robot localization and mapping with uncertainty using scale-invariant landmarks*. International Journal of Robotics Research, Vol. 21, No. 8 (2002) 735-758.

[9] S. Se, D. Lowe, J. Little. *Vision-based mapping with backward correction*. In Proceedings of IROS'02: IEEE/RSJ International Conference on Intelligent Robots and Systems (2002).

[10] S. Thrun, W. Burgard, D. Fox. *A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping*. In Proceedigns of ICRA'00: IEEE International Conference on Robotics and Automation (2000).

[11] Y. Liu, R. Emery, D. Chakrabarti, W. Bungard, and S. Thrun. *Using EM tlo learn 3D models with mobile robots*. In Proceedings of ICML'01: International Conference on Machine Learning (2001).

[12] C. Martin and S. Thrun. *Real-time acquisition of compact volumetric maps with mobile robots*. In Proceedings of ICRA'02: IEEE International Conference on Robotics and Automation (2002).

[13] D. Hähnel, W. Burgard, and S. Thrun. *Learning compact 3D models of indoor and outdoor environments with a mobile robot*. Robotics and Autonomous Systems, 44 (2003) 15-27.

[14] H. Surmann, A. Nücher, J. Hertzberg. *An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments*. Robotics and Autonomous Systems 45 (2003) 181-198.

[15] S.F. El-Hakim, P. Boulanger, F. Blais, and J.-A. Beraldin. *Sensor-based creation of indoor virtual environment models.* In Proceedings of VSMM'97: IEEE Confernce on Virtual Systems and Multimedia (1997).

[16] A. Gueorguiev, P. Allen, E. Gold, and P. Blaer. *Design, architecture, and control of a mobile site modeling robot*. In Proceedings of ICRA'00: IEEE International Conference on Robotics and Automation (2000).

[17] P. Allen, I. Stamos, A. Gueorguiev, E. Gold, and P. Blaer. *AVENUE: Automated site modeling in urban environments*. In Proceedings of 3DIM2001: International Conference on 3D Digital Imaging and Modeling (2001).

[18] F. Lu, E. Milios. *Globally consistent range scan alignment for environment mapping*. Autonomous Robots 4(4) (1997) 333-349.

[19] P. J. Besl and N. D. McKay. *A method for registration of 3D shapes*. IEEE Transactions on Pattern Analysis and Machine Intellitence 14(2) (1992) 239-256.

[20] J. Luck, C. Little, and W. Hoff. *Registration of range data using a hybrid simulated annealing amd Iterative Closest Point*. In Proceediings of ICRA'00: IEEE International Conference on Robotics and Automation (2000).

[21] Z. Zhang. *Iterative point matching for registration of free-form curves and surfaces*. International Journal of Computer Vision 13(2) (1994) 119-152.

[22] M.A.Fishler, R.C. Bolles. *Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography*. Comunications of the ACM 24 (1981)381-395.

[23] J. Lengyel, M. Reichert, B. Donald, and D. Greenberg. *Real-time robot motion planning using rasterizing computer graphics hardware*. In Proceedings of SIGGRAPH (1990).

[24] J. Barraquand, B. Langlois, and J. Latombe. *Numerical potential field techniques for robot path planning*. IEEE Transactions on Systems, Man, and Cybernetics, 22(2) (1992) 224-241.

[25] D. Murray and C. Jennings.*Stereo vision based mapping and navigation for mobile robots*. In Proceedings of ICRA'97: International Conference on Robotics and Automation (1997).