3-D Modelling and Robot Localization from Visual and Range Data in Natural Scenes

Carlos Parra*, Rafael Murrieta-Cid**, Michel Devy, and Maurice Briot

LAAS-CNRS, 7 Av. du Colonel Roche, 31077 Toulouse Cédex 4,
France
{carlos, murrieta, michel, briot}@laas.fr

Abstract. This paper concerns the exploration of a natural environment by a mobile robot equipped with both a video camera and a range sensor (stereo or laser range finder); we focus on the interest of such a multisensory system to deal with the incremental construction of a global model of the environment and with the 3-D localization of the mobile robot. The 3-D segmentation of the range data provides a geometrical scene description: the regions issued from the segmentation step correspond either to the ground or to objects emerging from this ground (e.g. rocks, vegetations). The 3D boundaries of these regions can be projected on the video image, so that each one can be characterized and afterwards identified, by a probabilistic method, to obtain its nature (e.g. soil, rocks ...); the ground region can be over-segmented, adding visual information, such as the texture. During the robot motions, a slow and a fast processes are simultaneously executed; in the modelling process (currently 0.1Hz), a global landmark-based model is incrementally built and the robot situation can be estimated if some discriminant landmarks are selected from the detected objects in the range data; in the tracking process (currently 1Hz), selected landmarks are tracked in the visual data. The tracking results are used to simplify the matching between landmarks in the modelling process.

1 Introduction

This paper deals with perception functions required on an autonomous robot which must explore a natural environment without any a priori knowledge. From a sequence of range and video images acquired during the motion, the robot must incrementally build a model and correct its situation estimate.
The proposed approach is suitable for environments in which (1) the terrain is mostly flat, but can be made by several surfaces with different orientations (i.e. different areas with a rather horizontal ground, and slopes to connect these areas) and (2) objects (bulges or little depressions) can be distinguished from the ground.

* This research was funded by the PCP program (Colombia -COLCIENCIAS- and France -Foreign Office-)
** This research was funded by CONACyT, México

H.I. Christensen (Ed.): ICVS'99, LNCS 1542, pp. 450–468, 1999.

Our previous method [5] [4] dedicated to the exploration of such an environment, aimed to build an object-based model, considering only range data. An intensive evaluation of this method has shown that the main difficulty comes from the matching of objects perceived in multiple views acquired along the robot paths. From numerical features extracted from the model of the matched objects, the robot localization can be updated (correction of the estimated robot situation provided by internal sensors: odometry, compass, ...) and the local models extracted from the different views can be consistently fused in a global one. The global model was only a stochastic map in which the robot situation and the object features and the associated variance-covariance matrix were represented in a same reference frame; the reference frame can be defined for example as the first robot situation during the exploration task. Robot localization, fusion of matched objects and introduction of newly perceived objects are executed each time a local model is built from a newly acquired image [21]. If some mistakes occur in the object matchings, numerical errors are introduced in the global model and the robot situation can be lost.

In this paper, we focus on an improved modelling method, based on a multisensory cooperation; in order to make faster and more reliable the matching step, both range and visual data are used. Moreover, the global model has now several levels, like in [13]: a topological level gives the relationships between the different ground surfaces (connectivity graph); the model of each terrain area is a stochastic map which gives information only for the objects detected on this area: this map gives the position of these objects with respect to a local frame linked to the area (first robot situation when this area has been reached).

In the next section, an overview of our current method is presented. It involves a general function which performs the construction of a local model for the perceived scene; it will be detailed in section 3. The exploration task is executed by three different processes, which are detailed in the three following sections: the initialization process is executed only at the beginning or after the detection of an inconsistency by the modelling process; this last one is a slow loop (from 0.1 to 0.2 Hz according to the scene complexity and the available computer) from the acquisition of range and visual data to the global model updating; the tracking process is a fast loop (from 1 to 3 Hz), which require only the acquisition of an intensity image.

Then, several experiments on actual images acquired in lunar-like environment, are presented and analized.

2 The General Approach

We have described on figure 1 the relationships between the main representations built by our system, and the different processes which provide or update these representations.

A 3-D segmentation algorithm provides a synthetic description of the scene. Elements issued from the segmentation stage are then characterized and afterwards identified in order to obtain their nature (e.g. soil, rocks ...).

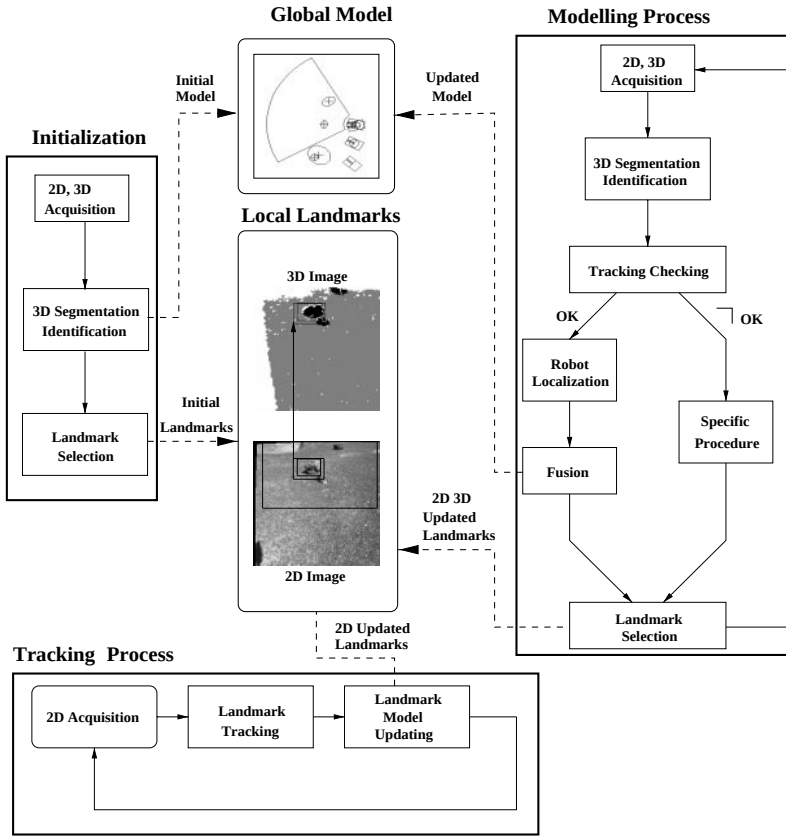


Fig. 1. The general approach

The nature of the elements (objects, ground) in the scene is obtained by comparing an attribute vector (computed from the shape and texture informations extracted from sensory data associated with this element) with a database. This database is function of the type of the environment. Here, we just have chosen 2 classes, which correspond to the principal elements in our environment: soil and rock. New classes inclusion as rocky soil and ground depressions (holes) are currently going on.

These phases allow us to obtain a local model of the scene. From this model, discriminant features can be extracted and pertinent objects for the localization tasks are selected as landmarks; according to some criteria which depend on higher decisional levels, one of these landmark is chosen as a tracked target; this same landmark could also be used as a goal for visual navigation. The tracking process exploits only a 2D image sequence in order to track the selected target while the robot is going forward. When it is required, the modelling process is

executed: a local model of the perceived scene is built; the robot localization is performed from matchings between landmarks extracted in this local model, and those previously merged in the global model; if the robot situation can be updated, the models of these matched landmarks are fused and new ones are added to the global model.

The matching problem of landmark's representation between different perceptions is solved by using the result of the tracking process. Moreover, some verifications between informations extracted from the 2D and 3D images, allow to check the coherence of the whole modelling results; especially, a tracking checker is based on the semantical labels added to the extracted objects by the identification function.

3 Local Scene Modelling

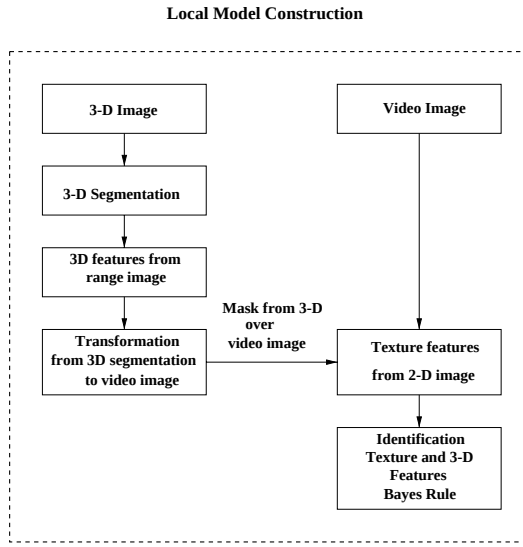


Fig. 2. The local model construction

The local model of the perceived scene is required in order to initialize the exploration task, and also to deal with the incremental construction of a global model of the environment.

The construction of this local model is performed from the acquisition of a 3D image by the range sensor, and of a 2D image from the video sensor (see figure 2), thanks to the following steps [18]:

– 3-D segmentation of the 3D image.

- Object characterization using the 3D and the 2D data.
- Object identification by a probabilistic method.

3.1 The 3-D Segmentation Algorithm

The main objective of the 3D segmentation step, is to extract the main scene components from the current 3D image. Two classes of components are considered: the ground and the obstacles. The ground corresponds to the surface on which the robot stands. This one is first identified, then the different obstacles are separated by a region segmentation of the 3D image. For more details about this segmentation algorithm, see [3].

To segment the ground, we must search in the scene the wider surface of uniform orientation which consequently corresponds to points having equivalent normal direction. This is mainly done by calculating a bi-variable histogram H which represents the number of points of the 3-D image having a given normal orientation, coded in spherical coordinates (θ, φ). The figure 3 (a) shows one video image; only correlated pixel by the stereovision module are displayed: white pixels correspond to occlusion or to distant 3-D points. The figure 3 (b) shows the range image (top view) and the figure 3 (c) presents the bi-variable histogram.

The predominance of the ground will appear as a peak in this bi-variable histogram even if the normals are noisy. This peak corresponds to points having normal direction close to the Z-axis of the robot frame; these points are essentially on the ground in front of the robot, but some other points can be linked to this peak: especially, the ones on the top of the obstacles. These points will be eliminated from the ground using the construction of a second histogram [3].

Once the ground regions have been extracted in the image, it remains the obstacle regions which could require a specific segmentation in order to isolate each obstacle. We make the assumption that an obstacle is a connected portion of matter emerging from the ground. Differents obstacles are separated by empty space which could be identified as depth discontinuities in the 3D image; these discontinuities are detected in a depth image, in which for each 3D point of the 3D image, the corresponding pixel value encodes the depth with respect to the sensor. Thus a classical derivative filter can be applied to obtain maxima of gradient corresponding to the depth discontinuities. Classical problems of edge closing are solved with a specific filter described in [3].

Finally the scene nature is deduced with a criterion based on:

- **number of peaks of H**: absence of a significative peak indicates that we could not easily differentiate the ground from the objects.
- **width of the peak**: if the peak is thin the ground is planar, otherwise the ground is very curved.
- **the mean error** caused by the approximation of the ground region with a given surface (plane or other shapes: paraboloid ...): the smaller the error, the more even is the ground.

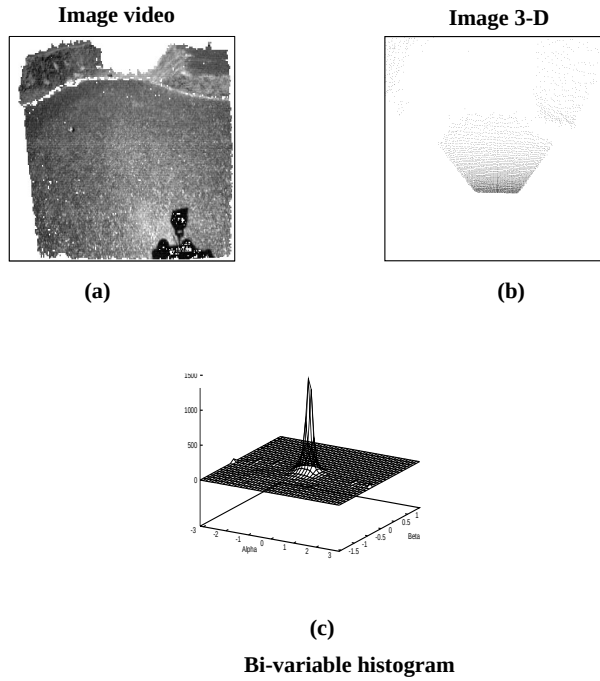


Fig. 3. (a) Video image (b) 3-D image (c) Bi-variable histogram

– **mean distance** of the 3D points along a common discontinuity line between two obstacle regions. If this distance is small the two obstacles are close to each other; they are grouped in a cluster, so that the mobile robot cannot move between them for example but, could easily perceive together these objects in multiple views.

The object-based representation we present in this paper, is only suitable when a peak is detected in the H histogram; in the other situation, a discrete representation (Digital Elevation Map) will be used in order to describe a ground region in an uneven terrain (see [11]). But in the more general situation, due to the high resolution of the range sensor close to the robot, our method detects only one peak, which corresponds to the ground area on which the robot is currently moving; a second peak is detected only when the robot comes near a slope discontinuity of the ground.

3.2 Object Characterization

Each object of the scene is characterized by an attribute vector: the object attributes correspond either to 3D features extracted from the 3D image or to

its texture extracted from the 2D image. The 3D features correspond to the statistical mean and the standard deviation of the distances from the 3-D points of the object, with respect to the plane which approximates the ground area from which this object is emerging.

We want also to associate intensity attributes to an object extracted from the 3D image; this object creates a 2D region in the intensity image acquired at the same time than the 3D one. On our LAMA robot, the 3D image is provided by a stereovision algorithm [12]; for the 2D image, two different sensor configurations have been considered:

- either we are only interested by the texture information, and the stereo images have a sufficient resolution. The left stereo image provides the 2D image on which the texture information will be computed; the indexes between the 3D points and the 2D points are the same, so that the object region extracted from the 3D image is directly mapped on the 2D image.
- or we want to take advantage of a high-resolution camera, or of a color camera. In such a case, the 2D image is provided by a specific camera, and an calibration procedure must be executed off line, in order to estimate the relative position between the 2D and the 3D sensors; the 2D region created by an object extracted from the 3D image, is provided by the projection on the 2D image of the 3D border line of the object.

The texture operators are based on the sum and difference histograms, this type of texture measure is an alternative to the usual co-occurrence matrices used for texture analysis. The sum and difference histograms used conjointly are nearly as powerful as co-occurrence matrices for texture discrimination. This texture analysis method requires less computation time and less memory requirements than the conventional spatial grey level dependence method.

For a given region of a video image $I(x, y) \in [0, 255]$, the sum and difference histograms are defined as [23]:

$$h_s(i) = Card(i = I(x, y) + I(x + \delta x, y + \delta y)) \quad i \in [0, 510]$$
$$h_d(j) = Card(j = \mid I(x, y) - I(x + \delta x, y + \delta y) \mid) \; j \in [0, 255]$$

The relative displacement $(\delta x, \delta y)$ may be equivalently characterized by a distance in radial units and an angle θ with respect to the image line orientation: this displacement must be chosen so that the computed texture attributes allow to discriminate the interesting classes; for our problem, we have chosen: $\delta x = \delta y = 1$. Sum and difference images can be built so that, for all pixel $I(x, y)$ of the input image, we have:

$$I_s(x, y) = I(x, y) + I(x + \delta x, y + \delta y)$$
$$I_d(x, y) = \mid I(x, y) - I(x + \delta x, y + \delta y) \mid$$

Furthermore, normalized sum and difference histograms can be computed for selected regions of the image, so that:

$$H_s(i) = \frac{Card(i = I_s(x,y))}{m} \; H_s(i) \in [0, 1]$$
$$H_d(j) = \frac{Card(j = I_d(x,y))}{m} \; H_d(j) \in [0, 1]$$

where m is the number of points belonging to the considered region.

Texture Feature	Equation
Mean	$\mu = \frac{1}{2}\sum_i i \cdot \hat{P}_{s(i)}$
Variance	$\frac{1}{2}(\sum_i (i-2\mu)^2 \cdot \hat{P}_{s(i)} + \sum_j j^2 \cdot \hat{P}_{d(j)})$
Energy	$\sum_i \hat{P}^2_{s(i)} \cdot \sum_j \hat{P}^2_{d(j)}$
Entropy	$-\sum_i \hat{P}_{s(i)} \cdot \log \hat{P}_{s(i)} - \sum_j \hat{P}_{d(j)} \cdot \log \hat{P}_{d(j)}$
Contrast	$\sum_j j^2 \cdot \hat{P}_{d(j)}$
Homogeneity	$\frac{1}{1+j^2}\sum_j \cdot \hat{P}_{d(j)}$

Table 1. Texture features computed from sum and difference histograms

These normalized histograms can be interpreted as a probability. $\hat{P}_{s(i)} = H_s(i)$ is the estimated probability that the sum of the pixels $I(x, y)$ and $I(x + \delta x, y + \delta y)$ will have the value i. And $\hat{P}_{d(j)} = H_d(j)$ is the estimated probability that the absolute difference of the pixels $I(x, y)$ and $I(x + \delta x, y + \delta y)$ will have value j.

In this way we obtain a probabilistic characterization of the spatial organization of the image, based on neighborhood analysis. Statistical information can be extracted from these histograms. We have used 6 texture features computed from the sum and difference histograms, these features are defined in Table 1.

3.3 Object Identification

The nature (class) of an object perceived in the scene is obtained by comparing its attribute vector (computed from the 3D features and from the texture) with a database composed by different classes, issued from a learning step executed off line.

This identification phase allows us to get a probabilistic estimation about the object nature. The label associated to an object, will be exploited in order to detect possible incoherences at two levels:

- at first, in the modelling process, a 3D segmentation error will be detected if the extracted objects cannot be labelled by the identification function.
- then, in the tracking process, the nature of the landmark could be used in addition to the partial Hausdorff distance to detect possible tracking errors or drifts.

A Bayesian classification is used in order to estimate the class membership for each object. The Bayesian rule is defined as [1]:

$$P(C_i \mid X) = \frac{P(X \mid C_i)P(C_i)}{\sum_{i=1}^{n} P(X \mid C_i)P(C_i)}$$

where:

- $P(C_i)$ is the *a priori* probability that an object belongs to the class (C_i).
- $P(X \mid C_i)$ is the class conditional probability that the object attribute is X, given that it belongs to class C_i.
- $P(C_i \mid X)$ is the *a posteriori* conditional probability that the object class membership is C_i, given that the object attribute is X.

We have assumed equal *a priori* probability. In this case the computation of the *a posteriori* probability $P(C_i \mid X)$ can be simplified and its value just depend on $P(X \mid C_i)$.

The value of $P(X \mid C_i)$ is estimated by using k-nearest neighbor method. It consists in computing for each class, the distance from the sample X (corresponding to the object to identify, whose coordinates are given by the vector of 3-D information and texture features) to $k - th$ nearest neighbor amongst the learned samples. So we have to compute only this distance (in common Euclidean distance) in order to evaluate $P(X \mid C_i)$. Finally the observation X will be assigned to the class C_i whose $k - th$ nearest neighbor to X is closest to X than for any other training class.

4 Initialization Phase

The initialization phase is composed by two main steps; at first, a local model is built from the first robot position in the environment; then, by using this first local model, a landmark is chosen amongst the objects detected in this first scene. This landmark will be used for several functions:

- it will support the first reference frame linked to the current area explored by the robot; so that, the initial robot situation in the environment must be easily computed.
- it will be the first tracked target in the 2D image sequence acquired during the next robot motion (tracking process: fast loop); if in the higher level of the decisional system, a visual navigation is chosen as a way to define the robot motions during the exploration task, this same process will be also in charge of generating commands for the mobile robot and for the pan and tilt platform on which the cameras are mounted.
- it will be detected again in the next 3D image acquired in the modelling process, so that the robot situation could be easily updated, as this landmark supports the reference frame of the explored area.

Moreover, the first local model allows to initialize the global model which will be upgraded by the incremental fusion of the local models built from the next

3D acquisitions. Hereafter, the automatic procedure for the landmark selection is presented.

The local model of the first scene (obtained from the 3-D segmentation and identification phases) is used to select automatically an appropriated landmark, from a utility estimation based on both its nature and shape [19].

Localization based on environment features improves the autonomy of the robot. A landmark is defined first as a remarkable object, which should have some properties that will be exploited for the robot localization or for the visual navigation, for example:

- **Discrimination**. A landmark should be easy to differentiate from other surrounding objects.
- **Accuracy**. A landmark must be accurate enough so that it can allow to reduce the uncertainty on the robot situation, because it will be used to deal with the robot localization.

Landmarks in indoor environments correspond to structured scene components, such as walls, corners, doors, etc. In outdoor natural scenes, landmarks are less structured: we have proposed several solutions like maxima of curvature on border lines [8], maxima of elevation on the terrain [11] or on extracted objects [4].

In previous work we have defined a landmark as a little bulge, typically a natural object emerging from a rather flat ground (e.g. a rock); only the elevation peak of such an object has been considered as a numerical attribute useful for the localization purpose. A realistic uncertainty model has been proposed for these peaks, so that the peak uncertainty is function of the rock sharpness, of the sensor noise and of the distance from the robot.

In a segmented 3D image, a bulge is selected as candidate landmark if:

1. It is not occluded by another object. If an object is occluded, it will be both difficult to find it in the following images and to have a good estimate on its top.
2. Its topmost point is accurate. This is function of the sensor noise, resolution and object top shape.
3. It must be in "ground contact".

These criteria are used so that only some objects extracted from an image are selected as landmarks. The most accurate one (or the more significative landmark cluster in cluttered scenes) is then selected in order to support the reference frame of the first explored area. Moreover, a specific landmark must be defined as the next tracked target for the tracking process; different criteria, coming from higher decisional levels, could be used for this selection, for example:

- track the sharper or the higher object: it will be easier to detect and to match between successive images.
- track the more distant object from the robot, towards a given direction (visual navigation).

- track the object which maximizes a utility function, taking into account several criteria (active exploration).
- or, in a teleprogrammed system, track the object pointed on the 2D image by an operator.

At this time due to integration constraints, only one landmark can be tracked during the robot motion. We are currently thinking about a multi-tracking method.

5 The Tracking Process (Fast-Loop)

The target tracking problem has received a great deal of attention in the computer vision community over the last years. Several methods have been reported in the literature, and a variety of features have been proposed to perform the tracking [7,16,9].

Our method is able to track an object in an image sequence in the case of a sensor motion or of an object motion. This method is based on the assumption that the 3D motion of the sensor or the object can be characterized by using only a 2D representation. This 2D motion in the image can be decomposed into two parts:

- A 2D image motion (translation and rotation), corresponding to the change of the target's position in the image space.
- A 2D shape change, corresponding to a new aspect of the target.

The tracking is done using a comparison between an image and a model. The model and the image are binary elements extracted from a sequence of gray levels images using an edge detector similar to [6].

A partial Hausdorff distance is used as a resemblance measurement between the target model and its presumed position in an image.

Given two sets of points P and Q, the Hausdorff distance is defined as [20]:

$$H(P,Q) = \max(h(P,Q), h(Q,P))$$

where

$$h(P,Q) = \max_{p \in P} \min_{q \in Q} \| p - q \|$$

and $\| \cdot \|$ is a given distance between two points p and q. The function $h(P,Q)$ (distance from set P to Q) is a measure of the degree in which each point in P is near to some point in Q. The Hausdorff distance is the maximum among $h(P,Q)$ and $h(Q,P)$.

By computing the Hausdorff distance in this way we obtain the most mismatched point between the two shapes compared; consequently, it is very sensitive to the presence of any outlying points. For that reason it is often appropriate to use a more general rank order measure, which replaces the maximization operation with a rank operation. This measure (partial distance) is defined as [14]:

$$h_k = K^{th}_{p \in P} \min_{q \in Q} \| p - q \|$$

where $K^{th}_{p \in P} f(p)$ denotes the K^{-th} ranked value of $f(p)$ over the set P.

5.1 Finding the Model Position

The first task to be accomplished is to define the position of the model M_t in the next image I_{t+1} of the sequence. The search for the model in the image (or image's region) is done in some selected direction. We are using the unidirectional partial distance from the model to the image to achieve this first step.

The minimum value of $h_{k1}(M_t, I_{t+1})$ identifies the best "position" of M_t in I_{t+1}, under the action of some group of translations G. It is possible also to identify the set of translations of M_t such that $h_{k1}(M_t, I_{t+1})$ is no larger than some value τ, in this case there may be multiple translations that have essentially the same quality [15].

However, rather than computing the single translation giving the minimum distance or the set of translations, such that its correspond h_{k1} is no larger than τ, it is possible to find the first translation g, such that its associated h_{k1} is no larger than τ, for a given search direction.

Although the first translation which $h_{k1}(M_t, I_{t+1})$ associated is less than τ it is not necessarily the best one, whether τ is small, the translation g should be quite good. This is better than computing all the set of valuable translation, whereas the computing time is significantly smaller.

5.2 Building the New Model

Having found the position of the model M_t in the next image I_{t+1} of the sequence, we now have to build the new model M_{t+1} by determining which pixels of the image I_{t+1} are part of this new model.

The model is updated by using the unidirectional partial distance from the image to the model as a criterion for selecting the subset of images points I_{t+1} that belong to M_{t+1}. The new model is defined as:

$$M_{t+1} = \{q \in I_{t+1} \mid h_{k2}(I_{t+1}, g(M_t)) < \delta\}$$

Where $g(M_t)$ is the model at the time t under the action of the translation g, and δ controls the degree to which the method is able to track objects that change shape.

In order to allow models that may be changing in size, this size is increased whenever there is a significant number of nonzero pixels near the boundary and is decreased in the contrary case. The model's position is improved according to the position where the model's boundary was defined.

The initial model is obtained by using the local model of the scene previously computed. With this initial model the tracking begins, finding progressively the new position of the target and updating the model. The tracking of the model is successful if:

$$k1 > fM \mid h_{k1}(M_t, I_{t+1}) < \tau$$

and

$$k2 > fI \mid h_{k2}(I_{t+1}, g(M_t)) < \delta \ ,$$

in which fM is a fraction of the number total of points of the model M_t and fI is a fraction of image's point of I_{t+1} superimposed on $g(M_t)$.

5.3 Our Contributions over the General Tracking Method

Several previous works have used the Hausdorff distance as a resemblance measure in order to track an object [15,10]. This section enumerates some of the extensions that we have made over the general method [17].

- Firstly, we are using an automatic identification method in order to select the initial model. This method uses several attributes of the image such as texture and 3-D shape.
- Only a small region of the image is examined to obtain the new target position, as opposed to the entire image. In this manner, the computation time is decreased significantly. The idea behind a local exploration of the image is that if the execution of the code is quick enough, the new target position will then lie within a vicinity of the previous one. We are trading the capacity to find the target in the whole image in order to increase the speed of computation of the new position and shape of the model. In this way, the robustness of the method is increased to handle target deformations, since it is less likely that the shape of the model will change significantly in a small δt. In addition, this technique allows the program to report the target's location to any external systems with a higher frequency (for an application see [2]).
- Instead of computing the set of translations of M_t, such that $h_{k1}(M_t, I_{t+1})$ is no larger than some value τ, we are finding the first translation whose $h_{k1}(M_t, I_{t+1})$ is less than τ. This strategy significantly decreases the computational time.

5.4 Experimental Results: Tracking

The tracking method was implemented in C on a real-time operating system (Power-PC), the computation running time is dependent on the region size examined to obtain the new target position. For sequences the code is capable of processing a frame in about 0.25 seconds. In this case only a small region of the image is examined given that the new target position will lie within a vicinity of the previous one. Processing includes, edge detection, target localization, and model updating for a video image of (256x256 pixels).

Figures 4 show the tracking process. Figure 4 a) shows initial target selection, in this case the user specifies a rectangle in the frame that contains the target. An automatic landmark (target) selection is possible by using the local model of the scene. Figures 4 b), c), d), and e) show the tracking of a rock through an image sequence. The rock chosen as target is marked in the figure with a boundary box. Another boundary box is used to delineate the improved target position after the model updating. In these images the region being examined is the whole image, the objective is to show the capacity of the method to identify a rock among the set of objects.

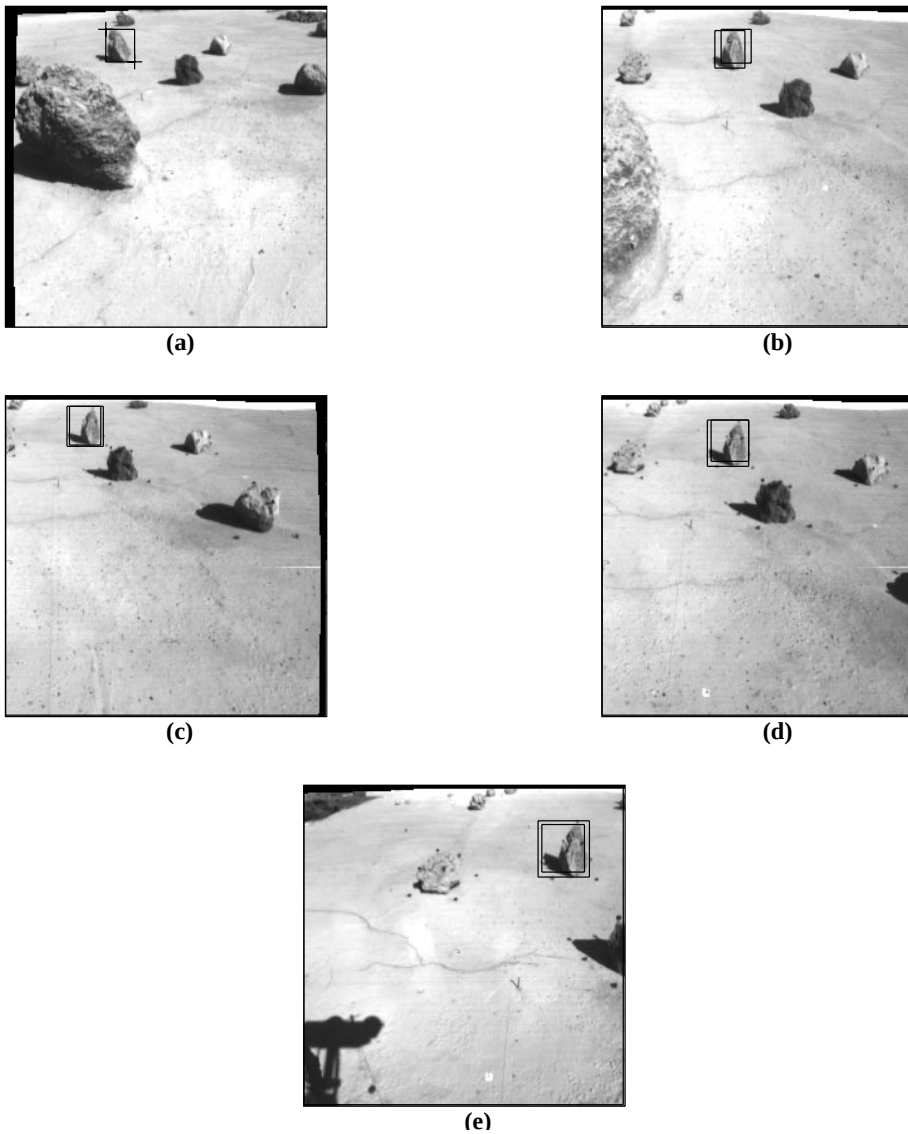


Fig. 4. Visual tracking

6 The Modelling Process (Slow Loop)

The local models extracted from the acquired 3D images are fused in order to build a global model in an incremental way. After each 3D acquisition, a local model is firstly built from the 3D image, by the use of the method described in section 3. Then the global model must be updated by merging it with the local one; this fusion function allows to improve the robot estimate position and attitude [22] [21].

6.1 Robot Localization and Global Model Fusion

The modelling process has an estimate of the robot situation provided by internal sensors (on the LAMA robot: odometers and inclinometers). This estimate may be quite inaccurate, and moreover systematically implies cumulative errors. The robot situation is represented by an uncertain vector $(x, y, z, \theta, \phi, \psi)$; the estimated errors is described by a variance-covariance matrix. When these errors become too large, the robot must correct its situation estimate by using other perceptual data; we do *not* take advantage of any *a priori* knowledge, such as artificial beacons or landmark positions, nor of external positionning systems, such as GPS. The self-localization function requires the registration of local models built at successive robot situations.

The global model has two main components; the first one describes the topological relationships between the detected ground areas; the second one contains the perceived informations for each area. The topological model is a connectivity graph between the detected areas (a node for each area, an edge between two connected areas). In this paper, we focus only on the knowledge extracted for a given area: the list of objects detected on this area, the ground model, and the list of the different robot positions when it has explored this area.

The global model construction requires the matching of several landmarks extracted in the local model and already known in the current global model. This problem has been solved using only the 3D images [4], but the proposed method was very unreliable in cluttered scenes (too many bad matchings between landmarks perceived on multiple views). Now, the matching problem is solved by using the visual tracking process. The landmark selected as the target at the previous iteration of the modelling process, has been tracked in the sequence of 2D images acquired since then. The result of the tracking process is checked, so that two situations may occur:

- in the local model built from the current position, we find an object extracted from the 3D image, which can be mapped on the region of the traked target in the corresponding 2D image. If the label given by the identification function to this region, is the same than the label of the target, then the tracking result is valid and the tracked landmark gives a first good matching from which other ones can be easily deduced.
- if some incoherences are detected (no mapping between an extracted 3D object and the 2D tracked region, no correspondance between the current

label of the tracked region and the previous one), then some specific procedure must be executed. At this time, as soon as no matchings can be found between the current local and global models, a new area is open: it means that the initialization procedure is executed again in order to select the best landmark in the local model as the new reference for the further iterations.

When matchings between landmarks can be found, the fusion functions have been presented in [4]. The main characteristics of our method is the uncertainty representation; at instant k, a random vector $\mathbf{X}_k = [\mathbf{x}_r^T \ \mathbf{x}_1^T \ ... \ \mathbf{x}_N^T]_k^T$ and the associated variance-covariance matrix represent the current state of the environment. It includes the current robot's situation and the numerical attributes of the landmark features, expressed with respect to a global reference frame. Robot situation and landmark feature updates are done using an Extended Kalman Filter (EKF).

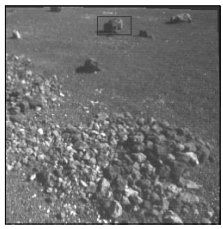
6.2 Experimental Results: Modelling

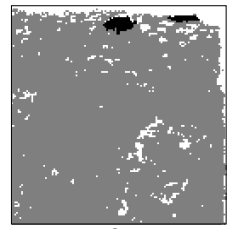
The figure 5 shows a partial result of the exploration task, involving concurrently the modelling and the tracking processes. Figure 5 I.a shows the video image, Figure 5 I.b presents the 3-D image segmentation and classification, two grey levels are used to label the classes (rocks and soil). Figure 5 I.c shows the first estimation of the robot position. A boundary box indicates the selected landmark (see figure 5 I.a). This one was automatically chosen by using the local model. The selection was done by taking into account 3-D shape and nature of the landmark.

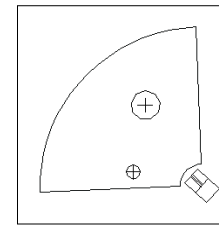
Figures 5 II and 5 III show the tracking of the landmark, which is marked in the figure with a boundary box. Another larger boundary box is used to delineate the region of examination.

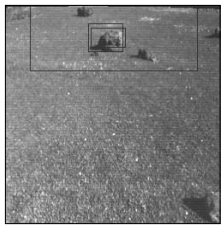
Figure 5 IV.a presents the next image of the sequence, figure 5 IV.b shows the 3-D segmentation and identification phases used to build the local model. The visual tracking is employed here to solve the matching problem of landmark's representation between the different perceptions. Figure 5 IV.c presents the current robot localization, the local model building at this time is merged to the global one. In this simple example, the global model contains only one ground area with a list of three detected landmarks and a list of two robot positions.

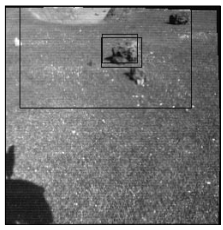
The target tracking process goes on in the next images of the sequence (see figure 5 V and figure 5 VI.a). The robot motion between the image V and VI.a was too important, so the aspect and position of the target changes a great deal; it occurs a tracking error (see the in figure 5 VI.b, the window around the presumed tracked target). A new local model is built at this time (figure 5 VI.b). The coherence of the both processes (local model construction and target tracking) is checked by using the nature of the landmark. As the system knows that the target is a rock, this one is able to detect the tracking process mistake given that the model of the landmark (target) belongs to the class soil.


I.a

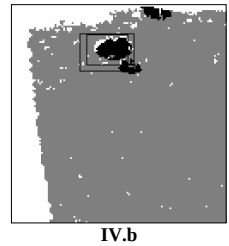

I.b

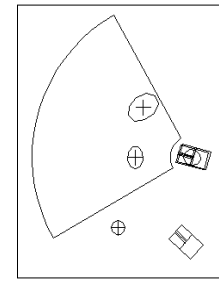

I.c

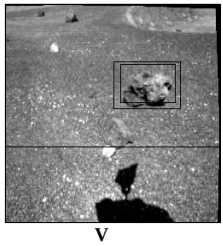

II

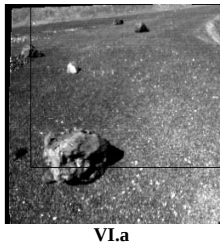

III


IV.a


IV.b


IV.c


V


VI.a

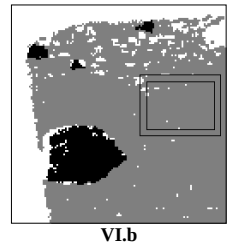

VI.b

Fig. 5. 3-D robot localization

7 Conclusion and Future Work

The work presented in this paper concerns the environment representation and the localization of a mobile robot which navigates on a rather flat ground in a planetary environment.

A local model of the environment is constructed in several phases:

- region extraction: firstly, the 3-D segmentation gives a synthetic representation of the environment.
- object characterization: each object of the scene is characterized by using 3-D features and its texture. Having done the segmentation both texture and 3-D features are used to characterize and to identify the objects. In this phase, texture is taken into account to profit from its power of discrimination. The texture attributes are computed from regions issued from the 3D segmentation, which commonly give more discriminant informations than the features obtained from an arbitrary division of the image.
- object identification: the nature of the elements (objects and ground) in the scene is obtained by comparing an attribute vector with a database composed by different classes, issued from a learning process.

The local model of the first scene is employed in order to select automatically an appropriate landmark. The matching problem of landmark's is solved by using a visual tracking process. The global model of the environment is updated at each perception and merged with the current local model. The current robot's situation and the numerical attributes of the landmark features are updated by using an Extended Kalman Filter (EKF).

Some possible extensions to this system are going on: firstly, we plan to study image preprocessors that would enhance the extraction of those image features that are appropriate to the tracking method. Second, we plan to include new classes (e.g. rocky soil and ground depressions) to improve the semantic description of the environment. Finally, we would also like to consider other environments such as natural terrestrial environments (e.g. forests or green areas). In this case, the color information could be taken into account, like we have proposed in [18,19].

References

1. H.C. Andrews. *Mathematicals Techniques in Pattern Recognition.* Wiley-Interscience, 1972.
2. C. Becker, H. González, J.-L. Latombe, and C. Tomasi. An intelligent observer. In *International Symposium on Experimental Robotics*, 1995.
3. S. Betgé-Brezetz, R. Chatila, and M. Devy. Natural Scene Understanding for Mobile Robot Navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, California, USA, May 1994.
4. S. Betgé-Brezetz, P. Hébert, R. Chatila, and M. Devy. Uncertain Map Making in Natural Environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, West Lafayette, USA, April 1996.

5. S. Betgé-Brezetz, R. Chatila, and M. Devy. Object-based modelling and localization in natural environments. In *Proc. IEEE International Conference on Robotics and Automation, Osaka (Japon)*, May 1995.
6. J. Canny. A computational approach to edge detection. *I.E.E.E. Transactions on Pattern Analysis and Machine Intelligence*, 8(6), 1986.
7. P. Delagnes, J. Benois, and D. Barba. Adjustable polygons: a novel active contour model for objects tracking on complex background. *Journal on communications*, 8(6), 1994.
8. M. Devy and C. Parra. 3D Scene Modelling and Curve-based Localization in Natural Environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'98)*, Leuven, Belgium, 1998.
9. Yue Du. A color projection for fast generic target tracking. In *International Conference on Intelligent Robots and Systems*, 1995.
10. M. Dubuisson and A. Jain. 2d matching of 3d moving objects in color outdoors scenes. In *I.E.E.E. Computer Society Conference on Computer Vision and Pattern Recognition*, june 1997.
11. P. Fillatreau, M. Devy, and R. Prajoux. Modelling of Unstructured Terrain and Feature Extraction using B -spline Surfaces. In *Proc. International Conference on Advanced Robotics (ICAR'93),Tokyo (Japan)*, November 1993.
12. H. Haddad, M. Khatib, S. Lacroix, and R. Chatila. Reactive navigation in outdoor environments using potential fields. In *International Conference on Robotics and Automation ICRA'98*, pages 1332–1237, may 1998.
13. H.Bulata and M.Devy. Incremental construction of a landmark-based and topological model of indoor environments by a mobile robot. In *Proc. 1996 IEEE International Conference on Robotics and Automation (ICRA'96), Minneapolis (USA)*, 1996.
14. D.P. Huttenlocher, A. Klanderman, and J. Rucklidge. Comparing images using the hausdorff distance. *I.E.E.E. Transactions on Pattern Analysis and Machine Intelligence*, 15(9), 1993.
15. D.P. Huttenlocher, W.J. Rucklidge, and J.J. Noh. Tracking non-rigid objects in complex scenes. In *Fourth International Conference on Computer Vision*, 1993.
16. S. Jiansho and C. Tomasi. Good features to track. In *Conference on Computer Vision and Pattern Recognition*, 1994.
17. R. Murrieta-Cid. Target tracking method based on a comparation between an image and a model. Technical Report Num. 97023, LAAS CNRS, written during a stay at Stanford University, Toulouse, France, 1997.
18. R. Murrieta-Cid. *Contribution au développement d'un système de Vision pour robot mobile d'extérieur.* PhD thesis, INPT, LAAS CNRS, Toulouse, France, November 1998.
19. R. Murrieta-Cid, M. Briot, and N. Vandapel. Landmark identification and tracking in natural environment. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'98)*, Victoria, Canada, 1998.
20. J. Serra. *Image analysis and mathematical morphology.* Academic Press, London, 1982.
21. R. C. Smith, M. Self, and P. Cheeseman. Estimating Uncertain Spatial Relationships in Robotics. *Autonomous Robot Vehicules*, pages 167–193, 1990.
22. K. T. Sutherland and B. Thompson. Localizing in Unstructured Environments: Dealing with the errors. *I.E.E.E. Transactions on Robotics and Automation*, 1994.
23. M. Unser. Sum and difference histograms for texture classification. *I.E.E.E. Transactions on Pattern Analysis and Machine Intelligence*, 1986.