Published in 5th International Symposium on Artificial Intelligence, Robotics and Automation in Space (I-SAIRAS 99), June 1999, Noordwijk, The Netherlands

ROVER SELF LOCALIZATION IN PLANETARY-LIKE ENVIRONMENTS

Simon Lacroix, Anthony Mallet and Raja Chatila
LAAS-CNRS
7 Av. du Colonel Roche - 31077 Toulouse Cedex 4 - France
Tel. +33 561 33 62 66 - Fax. +33 561 33 64 55
Simon.Lacroix@laas.fr, Anthony.Mallet@laas.fr, Raja.Chatila@laas.fr

Laurent Gallo
AEROSPATIALE
E/SCS/V - 2 Rue Béranger - 92320 Chatillon - France
Tel. +33 147 46 38 68 - Fax. +33 147 46 32 68
Laurent.Gallo@missiles.aerospatiale.fr

Abstract

The ability for a rover to localize itself with respect to its environment is a crucial issue to tackle autonomous long range navigation. In this paper, we first present and classify the various kind of functionalities a rover should be endowed with to estimate its position during long traverses. We then present a technique that relies on stereovision and pixel tracking to estimate the 6 parameters of the rover displacements, and discuss experimental results obtained with the robot Lama. The paper ends by a brief presentation of a complementary localization function, with respect to an object-based environment model built by the rover as it navigates.

1 Introduction

Future planetary exploration robots will have to explore, map or traverse larger and larger areas. This is a tremendous challenge for roboticists, that must conceive systems endowed with *autonomous long range navigation* capacities. Indeed, the various constraints related to planetary exploration (communication delays, poorly known unstructured terrain) void the possibility to efficiently teleoperate the machine.

At LAAS, we have tackled various aspects related to autonomous long range navigation in unstructured terrains for over ten years, and experimented some in realistic conditions [1, 2]. We are convinced that to efficiently achieve high level missions defined over a large scale of space and time, a certain degree of *deliberation* is necessary in order to anticipate events, take efficient decisions, and react adequately to unexpected events [3]. In particular, this robot ability to *plan* its activities calls for the building of various environment representations, at several levels of abstraction (topological maps, geometric maps, object representations...). For that purpose, an estimate of the robot position is required, and when executing long missions, sensors on board the lander are no longer helpful to compute it. A position estimate is not only necessary to build coherent environment models, it is also required to ensure that the given mission is successfully being achieved, or to servo motions along a defined trajectory: robot self-localization is actually one of the most important issue to tackle autonomous navigation.

The internal sensors of a robot being always subject to errors and drift, a lot of attention has been paid to exteroceptive data based position correction or estimation algorithms since the very beginning of mobile robotics. Basically, this problem is threefold: *(i)* the robot has to extract and associate relevant data or models from the gathered data, *(ii)* he has to process these associations to refine or estimate its position, *(iii)* and finally, he must be able to *actively* control its perception capacities in order to acquire the relevant data.

We focus in this paper on the first part of the problem, in the context of autonomous navigation in planetary-like environments. The problem is then very different from indoor environments, the context within which it has essentially been studied up to now: not only the internal sensors data are more noisy - the ground is seldom flat and smooth, but also the environment is not intrinsically structured, as compared to indoor environments where simple geometric primitives match fairly well the reality, and can therefore be “easily” associated from one point of view to the other.

The next section presents a tentative classification of the various position estimation techniques a planetary rover should be endowed with. Section 3 presents in details a technique that enables to estimate the robot motions with an excellent accuracy, using pixel tracking and stereovision. The following section sketch an approach we are currently developping to localize the robot over a long range, on the basis of an object-based environment representation built by the robot. A short discussion concerning the current trend in research related to robot environment modelling concludes the paper.

2 A tentative classification of exteroceptive localization techniques

The various techniques required to compute the robot position as it navigates range from inertial or odometry data integration to absolute localization respecting an initial model. In order to have a better understanding of the problem, we propose here to classify theses techniques into four functional categories:

1. *Motion estimation*: it consists in integrating data as a very high pace as the robot moves, similarly to proprioceptive localization[1], in order to estimate the parameters of elementary motions.

2. *Position refinement*: as with proprioceptive localization, exteroceptive motion estimation techniques generate cumulative errors. It is then necessary to rely on the association of elements in the environment (landmarks) perceived from quite different positions to refine the position estimate. The landmark matching problem is here easily solved thanks to the precise enough position estimate provided by the motion estimation technique.

3. *Position determination*: even when perceiving and memorizing landmarks, some errors on the position estimate cumulate over a long range of time and space (or after traveling a landmark-free area for instance). Such errors can reach very high values, so that when re-perceiving previously modeled landmarks, one can not rely on the current position estimation to match them. It then calls for *object recognition* to tackle the data association process.

4. *Absolute localization*: in this last category, we put all the techniques that aim at localizing the robot with respect to an initial global model of the environment (such as images or numerical terrain models derived from orbital imagery), a problem often referred to as the "drop-off problem" [4]). If descent imagery can be used to initially localize the lander [5], the problem of absolute localization still has to be tackled when roving over several kilometers.

There are actually five criteria that lead us to establish such a classification of the localization functions: *(i)* frequency of process activation, *(ii)* requirements on the precision of the initial robot position, *(iii)* volume of data required, *(iv)* necessity to control the data acquisition, and *(v)* level of abstraction of the processed data. For instance, the motion estimation functionality process a small amount of raw data at a very high frequency, without any control of the data acquisition, and may require a precise initial estimate of the motions (given by the proprioceptive sensors) in order to track and associate successfully the data. On the contrary, the absolute localization function is seldom triggered, requires a high level environment model built upon numerous data sets, for the construction of which data acquisition strategies have been determined, and by definition do not require any precise initial position estimate[2].

As one can see, the development of several different data processing and environment modeling algorithms is required to tackle the localization problem. All these algorithms are complementary, and provide position estimates with different characteristics: a model of each of these algorithms is required in order to filter the various position estimates into a consistent one, and to plan or trigger their activation.

3 Motion estimation using stereovision and pixel tracking

We present here an exteroceptive position estimation technique that is able to estimate the 6 parameters of the robot displacements in any kind of environments, provided it is textured enough so that pixel-based stereovision works well (thanks to progresses on cameras and algorithms, it is even the case for very smooth and flat terrains - the presence of no particular landmark is required). Referring to the classification presented in the former section, this technique is a *motion estimation* function. It is *passive*, in the sense that it do not calls for any data acquisition strategy: images are just used as fast as possible.

[1] We denote by "proprioceptive localization" all the algorithms that estimate the robot position using proprioceptive sensors - odometers, accelerometers, gyroscopes, inclinometers, etc.

[2] Note that among these criteria, the abstraction level of the data is actually dubious: we will indeed see in the next sections that we tackle the motion estimation and object recognition functionalities using very similar data (raw grey level images)

The algorithms therefore do not interfere with any other functionality that makes use of the stereo cameras (obstacle avoidance, map building).

3.1 Principle of the approach

The approach we developed and experimented could be called "exteroceptive dead-reckoning": it computes an estimate of the 6 displacement parameters between two stereo frames on the basis of a set of 3D point to 3D point matches, established by tracking the corresponding pixels in the image sequence acquired while the robot moves (figure 1). Depending on the time spent by stereovision and on the number of pixels to track, the tracking phase lasts a variable number of frames, which can be reduced to one.

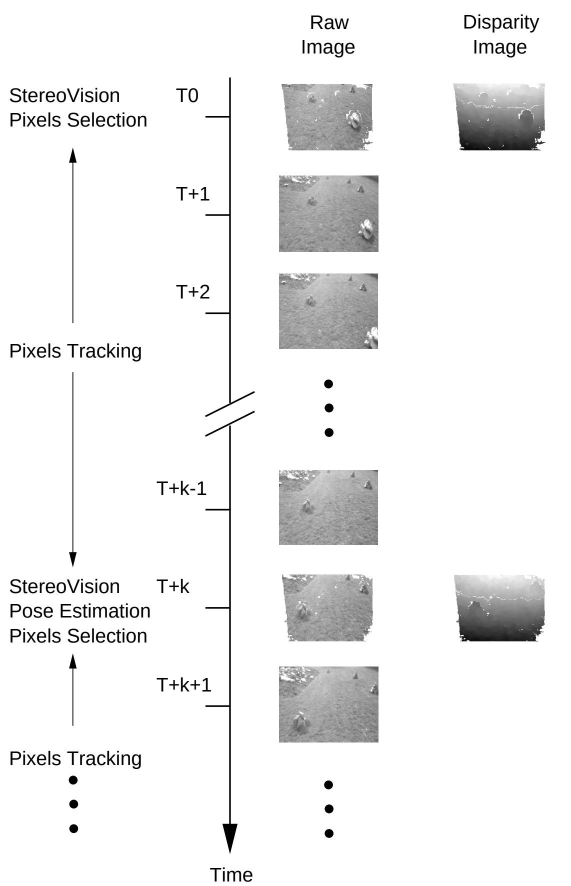

Figure 1: *Principle of the approach: at time T_0, a correlation algorithm computes a disparity image from a stereo pair, and a set of pixels to track is selected. Between T_1 and T_k, the selected pixels are tracked in the image sequence. After the determination of the disparity image at T_k, the set of 3D points correspondences $\{P_0^i, P_k^i\}$ established by the tracking phase is used to compute the displacement $Tr_{0\rightarrow k}$, and the process starts again.*

The principle of the approach is extremely simple, but paid we a lot of attention to the selection of the pixel to track: in order to avoid wrong correspondences, one must make sure that they can be faithfully tracked, and in order to have a precise estimation of the motion, one must choose pixels whose corresponding 3D point is known with a good accuracy. Pixel selection is done in three steps: an *a priori* selection is done on the basis of the stereo images (section 3.2); a model of the pixel tracking algorithm is used to discard the dubious pixels during the tracking phase (section 3.3); and finally an outlier rejection is performed when computing an estimation of displacement between two stereo frames (*a posteriori selection* - section 3.4).

3.2 Selection of the pixels to track

To initiate the process as a stereo frame comes up, one must select a set of pixels to be tracked. On one hand, one would like to track pixels whose corresponding 3D point is known with a good accuracy: this is done thanks to an error model of the pixel-based stereovision algorithm. On the other hand, one would like to select pixels that are likely to be successfully tracked in the forthcoming image sequence: this is done by studying the behavior of the auto-correlation function in the neighbor of the pixels of the image.

An error model for pixel correlation-based stereovision: A dense disparity image is produced from a pair of images thanks to a correlation-based pixel matching algorithm (we use the ZNCC correlation criteria or a Hamming distance computed on Census transformed images [6]). False matches are avoided thanks to a reverse correlation and to various thresholds defined on the correlation score curve (essentially on the value of the highest score, and on between this score and the second highest peak in the curve). To get quantitative informations on the precision of the computed disparity (and therefore on the coordinates of the 3D points), we studied a set of 100 images acquired from the same position. As in [7], it appeared that the distribution of the standard deviation on the disparity estimate can be well approximated by a Gaussian. Not surprisingly, the standard deviation on the depth increases quadratically with the depth[3]. A more interesting fact is that there is a *strong correlation* between the shape of the correlation curve around its peak and the standard deviation on the disparity: the sharper the peak, the more precise the disparity found. This correlation defines an *error model*, that is used during the correlation phase to estimate the error on the computed disparity (figure 2).

[3]This would actually be true if the standard deviations on the disparities were not dependent of the depth. In practice, further areas being less textured than closer ones, the disparity standard deviation increases with the depth. As a consequence, the depth standard deviation increases more than with the square of the depth

Figure 2: *A result of our stereovision algorithm: from left to right, original image (only correlated pixels are shown), disparity image, and standard deviation on the disparity estimated with our error model.*

However, there are matching errors that occur at the border between two regions of very different intensity values located at different depths (figure 3): as a consequence, the object shape in the disparity image is artificially growed of half the size of the correlation window. These errors, often referred to as "occluding contours artifacts" [8] can not be filtered out thanks to the thresholds on the correlation curve or to a blob filtering algorithm. Moreover, their estimated error tend to be very small: it is practically impossible to avoid the selection of such pixels considering only the stereovision algorithm model.

Figure 3: *False matches at the border of a rock: disparity image (left), and correlated pixels (right).*

Selecting good candidates for the tracking algorithm: Planetary environments being highly textured, simple area-based matching techniques are extremely efficient to track pixels in an image sequence (see section 3.3). However, due to noise in the image and the sampling performed by cameras, the tracking algorithm often eventually drifts: after a few image frames, tracked pixels do not correspond to the same terrain points than the points corresponding to the original pixels. This off course occur especially on smooth, low textured areas, but can also occur on highly textured areas: checking a simple threshold on the standard deviation on the grey levels of the correlation window is not sufficient to ensure that a pixel will be successfully tracked.

To avoid the selection of pixels in the image that are likely to drift during the tracking phase, we defined a measure other the image that represents how similar is a pixel to its neighbors. This measure is based on the computation of the correlation score of one pixel with each of its neighbors, using the same correlation score and window size as the tracking algorithm (auto-correlation). These scores define a correlation peak (a surface), and the shape of this peak indicates how different is one pixel from its neighbors: the sharper the peak, the more different are the neighbors from the pixel. We use the greatest value of the correlation scores found for the neighbors as an indicator of the sharpness of the peak, divided by the theoretical maximum correlation score.

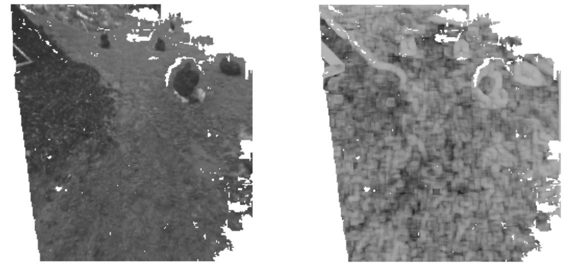

Figure 4: *Local similarity measure computed over a whole image. Left: original image, right: similarity measure encoded as grey levels. The darker pixels are good candidates for the tracking algorithm.*

Figure 4 presents a result of the computation of this measure over a whole image. One can note that the pixels corresponding to occluding contours are not good candidates for the tracking algorithm: indeed, in the two directions defined by the contour, the correlation windows are very similar. Finally, note that this measure gives an indicator related to the expected *precision* of the tracking algorithm for a pixel, but not related to the ambiguity (*certainty*): to evaluate an ambiguity measure would require the computation of correlation scores for a wide neighborhood, which is extremely time consuming.

Pixels selection: The set of candidate pixels to track is defined by applying thresholds on the depth standard deviation estimate of the 3D points and on the corresponding pixel similarity measure. The pixels that will actually be tracked are then randomly chosen among the remaining candidates.

3.3 Tracking pixels in an image sequence

Although the pixels to track have been carefully selected, some errors (drifts or false matches) can occur during the tracking phase. In order to avoid such errors, we tested various matching criteria (SSD, ZNCC, Census...) and various template updating strategies on several image sequences to determine the best ones.

Thanks to stereo image sequences, we can detect when a tracking algorithm is drifting by tracking "stereo-corresponding" pixels in the two images, and by checking that after the tracking phase, the returned pixels are still corresponding in the new stereo pair. However, tracking in parallel pixels in a stereo pair takes twice the time to track pixels in one image. We therefore used this possibility to check off-line the tracking algorithm with stereovision, to establish statistics on various tracking algorithms and with various correlation window sizes. This helped to determine the best matching score, template update strategy and optimal window size: we retained the

ZNCC correlation score computed over a $11x11$ window, and update the template by interpolating the target image around the sub-pixellic matching estimate and with the previous template. Moreover, it allowed us to easily determine the threshold values on the maximum correlation score and on the difference between the second highest peak in the surface, thresholds under which the algorithm is suspected to drift or to return a false match.

The tracking phase is done as follows: given a set of pixel to track and their corresponding 3D points defined on the stereo frame T_0, the search zone in the image acquired at time T_1 is centered around their predicted position, using the transformation $Tr_{T_0 \to T_1}$ provided by the robot internal sensors. The size of the search zone is determined according to the uncertainty on the estimated transformation. This prediction is important: it helps to focus the match search in a small area, and therefore reduces the probability to return a false match. Figure 5 shows the result of tracking a set of pixels in two images acquired from two positions distant of about 0.1 meter. One can see that most pixels have been successfully tracked.

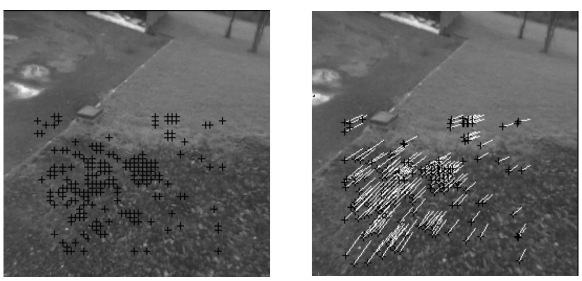

Figure 5: *Result of the tracking algorithm on a set of selected pixels. The pixels to track are displayed on the first image (left), and the found pixels on the following image are displayed on right.*

3.4 Estimation of the motions

At the date T_k, when a new stereo acquisition is performed, the pixels of the tracked set whose 3D coordinate estimate is now below a certain accuracy are discarded, and the remaining matches are used to compute a first estimate of the 3D transformation $T_{0 \to k}$, using a constrained least-square method [9]. On the basis of this first estimation, outliers are rejected from the set of matched 3D points, and a new estimate is computed. In our case, the outlier rejection is easy to achieve: indeed, thanks to the *a priori* selection phase and to the thresholds applied during the pixel tracking phase, most of the matched 3D points pairs are consistent.

Tracking pixels over several stereo frames: The obvious drawback of computing elementary motions only between two consecutive stereo frames is that the errors on the motion estimation cumulates over time, just as it happens when integrating the data of the robot's internal sensor. One way to reduce this errors is to use the possibility to track some pixels over several stereo frames: it allows to determine various displacements parameters every time a stereo image comes up (figure 6).

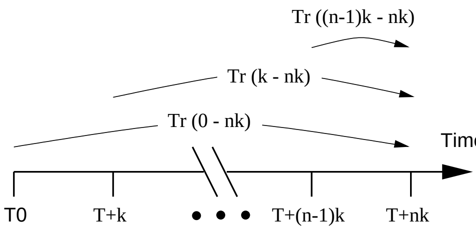

Figure 6: *Several displacements between stereo frames can be estimated*

One could imagine to combine these various displacements estimations using a stochastic filtering technique: this would require the precise knowledge of the uncertainties on every displacement estimation, which is not obvious to obtain. We solved this problem by computing a least square estimation for the whole set of possible 3D points matches in a recursive way: once a position corresponding to time T_{nk} is estimated, all the former 3D points coordinates are expressed in this position. At the frame $T_{(n+1)k}$, the matched 3D points are duplicated in order to generate all the possible association sets $\bigcup_{i=0}^{i=n} \{P_{ik} \leftrightarrow P_{(n+1)k}\}$, and the constrained least square method is applied on the whole associations.

The *a priori* pixel selection, performed every time a stereo frame is produced, is then only done to replace the pixels that have been lost (rejected during the tracking phase or as outliers) during the previous stereo-to-stereo cycle.

3.5 Functional architecture

We are currently integrating all the functionalities required by our approach on board the robot Lama. Figure 7 presents the necessary functional modules (integrated under the real-time operating system VxWorks thanks to GenoM, a software tool developed in our research group to specify and integrate libraries [10]), and the connections between them.

3.6 First experimental results

We have tested the approach with the robot Lama[4](figure 8), and established comparisons with position records obtained with a differential phase GPS localization system.

[4]Lama that is currently lent to us by Alcatel Space Industries

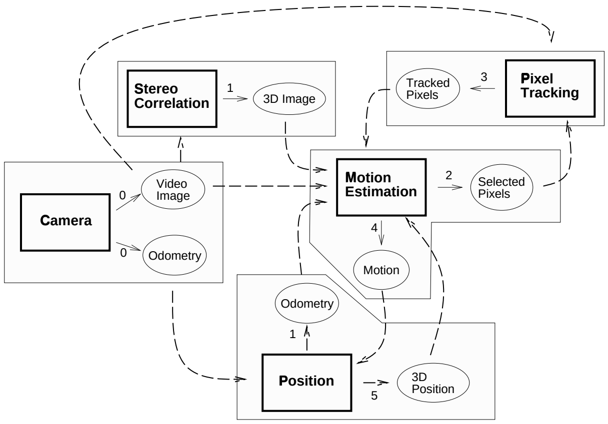

Figure 7: *Functional architecture of the motion estimation technique. Functional modules are displayed in bold rectangular boxes, and exported data are displayed in ovals. The numbers indicates the order of data production.*

Figure 8: *The robot Lama, a Marsokhod robot built by VNII Transmach and equipped at LAAS*

The first results are very promising: on some translations of several tens of centimeters, the displacement estimated by the algorithms was close up to 1% to the GPS positions, ie. as precise as the (up to now, we only characterized the translations). We are currently establishing thorough statistics, in order to precisely qualify the precision of the technique. On longer motions, that corresponds to several tens of stereo frames, it appeared clearly that there is a great advantage in tracking pixels over several stereo frames. However, we are not satisfied by the least square estimation algorithm: a Kalman filtering would surely do a better job, all the more since the uncertainty on the 3D points coordinates are well none.

4 Toward unstructured object recognition

We briefly present here a new technique we are currently working on, that allows to identify and register previously perceived objects. It can therefore satisfy both the position refinement and determination processes (section 2). Historically, the first attempts to solve these problems relied on analytical objects models (such as superquadrics for instance). The inadequacy of such models to unstructured objects lead us to study deformable meshes. However if these techniques are well suited for very precise geometric data (such as in medical imaging), they remain useless for robot navigation, where the data range several meters and are much more noisy. We think the recent advances in image registration may be successfully adapted to our problem: instead of aiming at building concise and precise models, these techniques tend to solve the data association problems using either global invariant features [11] or a set of local invariant features [12] determined in the images.

The principle of the method we propose is the following: it consists in building a database of object images (referred to as "aspects") as the robot navigates. Instead of computing local invariants for these pixels, as it is usually the case when indexing images, we make use of the 3D informations produced by stereovision for all the pixels to predict the object aspect for a *constant* camera distance and orientation[5]. This relaxes the need to compute invariants with respect to image scale and orientation. On this "projected" aspect, a set of discriminant pixels is marked using the autocorrelation function presented in section 3.2 (or using an Harris detector for instance). During the visual aspect database construction, a deformable mesh is determined on the basis of the various 3D points sets. No mesh registration is done: we assume the precision of the robot position given by the motion estimation technique is precise enough to build a mesh that is roughly consistant.

When perceiving an object after a while or a long distance travelled, the problem is to determine whether it has already been perceived or not, and if yes, to determine the robot location with respect to the memorized positions corresponding to aspects. This is done according to the following procedure:

1. The first phase consists in using global attributes to select among all the candidates aspects the ones that are most likely to be matched with the newly perceived object. These attributes are coarse geometric informations (such as the estimate of the object's volume and inertia moments derived from the

[5] The orientation of the camera with respect to the gravity vector is faithfully provided by inclinometers or the motion estimation technique.

mesh), and global photometric informations (texture for instance).

2. The second phase consists in selecting the aspect among the remaining candidates that resembles the most the current aspect. This is done on the basis of the current aspect prediction for the constant camera distance and orientation chosen during the database building, and by evaluating simple correlation scores (as in the motion estimation technique) for discriminant pixels provided by the autocorrelation function. The originality of the method rely here on the use of the 3D data to project the current aspect to a viewpoint as close as possible to the ones stored in the database, which relaxes the need to compute invariants. Up to now, the best candidate is only chosen using a measure defined on the correlation score for all the marked pixels. Geometric constraints between the marked pixels would probably be helpful when dealing with a large database: the problem is similar to primitive based object recognition techniques.

3. The last phase consists in determining the robot pose with respect to the matching aspect, using a 3D points set of correspondences, as in the motion estimation technique.

One of the critical point of the method is the ability to segment the objects in the data. We have only considered the easy case of rocks lying on a rather flat ground, and developed a simple object detection procedure on the disparity image. It relies on the possibility to quickly compute a virtual disparity image that corresponds to a theoretical flat ground, using the estimate of the robot attitude. A difference between this predicted image and the perceived disparity image exhibits the parts that are above the ground: a simple threshold on this difference lead to a "blob image", each blob corresponding to a potential object (similar simple segmentation techniques can be applied on the 3D points image). However, one of the advantage of our matching method is that is do not require a faithful segmentation.

5 Conclusions

Rover self-localization is an extremely important issue to tackle in order to endow a robot with autonomous long range navigation capacities. In this paper, we have discussed the various kind of functionalities to develop in order to solve this problem. These functionalities require various data processing and environment modeling algorithms, and may require the determination of data acquisition strategies.

We have presented an approach that is able to estimate elementary robot motions on the basis of stereovision, without building any environment model, and we have sketched an object modeling approach that can satisfy both the position refinement and determination processes over a long range.

Most investigations concerning localization in outdoor environments relied mainly on geometric characteristics: on rough terrains, a *digital map elevation* is directly used to feed an iconic matching procedure [13, 14], or geometric features extracted from the model are matched [15]. On rather flat terrains where obstacles are easily segmented, some techniques relying on geometric obstacle models have been proposed [16, 17]. However, if geometry is an essential feature to build environment models and indispensable to compute positions, the errors in the data and the models makes the association algorithms very fragile.

The approaches we presented rely essentially on image data, rather than on geometric data. Indeed, the increasing ability to store and rapidly process a large amount of data, due to performances progresses of the computers, lead us to develop techniques that make a strong use of raw image data. As it seems to be the current trend in the perception community, we tend to give up "reconstructionist" approaches that aim at building a concise representation. One can see that if geometry remains an indispensable feature, its role is strongly diminishing in the data association processes.

References

[1] R. Chatila and S. Lacroix. A case study in machine intelligence: Adaptive autonomous space rovers. In A. Zelinsky, editor, *International Conference on Field and Service Robotics, Canberra (Australia)*, number XII. Springer-Verlag, July 1998.

[2] S. Lacroix, R. Chatila, S. Fleury, M. Herrb, and T. Siméon. Autonomous navigation in outdoor environment : Adaptive approach and experiments. In *Proceedings of the IEEE International Conference on Robotics and Automation, San Diego, California (USA)*, May 1994.

[3] R. Chatila. Deliberation and reactivity in autonomous mobile robots. *Robotics and Autonomous Systems*, 16(2-4):197–211, 1995.

[4] F. Cozman and E. Krotkov. Automatic mountain detection and pose estimation for teleoperation of lunar rovers. In *International Symposium on Experimental Robotics, Barcelona (Spain)*, pages 164–172, June 1997.

[5] L. Matthies, C. Olson, G. Tharp, and S. Laubagh. Visual localization methods for mars rovers using lander, rover and descent imagery. In *International Symposium on Artificial Intelligence, Robotics and Automation in Space, Tokyo (Japan)*, pages 413–418, July 1997.

[6] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *Third European Conference on Computer Vision, Stockholm (Sweden)*, May 1994.

[7] L. Matthies. Toward stochastic modeling of obstacle detectability in passive stereo range imagery. In *IEEE International Conference on Computer Vision and Pattern Recognition, Champaign, Illinois (USA)*, pages 765–768, 1992.

[8] R. Sara and R. Bajcsy. On occluding contour artifacts in stereo vision. In *IEEE Conference on Computer Vision and Pattern Recognition, Puerto Rico*, pages 852–857, June 1997.

[9] R. Haralick, H. Joo, C.-N. Lee, X. Zhuang, V.G. Vaidya, and M.B. Kim. Pose estimation from corresponding point data. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(6):1426–1446, Nov/Dec 1989.

[10] R. Alami, R. Chatila, S. Fleury, M. Ghallab, and F. Ingrand. An architecture for autonomy. *Special Issue of the International Journal of Robotics Research on Integrated Architectures for Robot Control and Programming*, 17(4):315–337, April 1998. Rapport LAAS N97352, Septembre 1997, 46p.

[11] B. Schiele and J. Crowley. Object recognition using multidimensional receptive field histograms. In *Fourth European Conference on Computer Vision, Cambridge, UK*, April 1996.

[12] C. Schmid and R. Mohr. Local greyvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5), May 1997.

[13] I.S. Kweon and T. Kanade. High-resolution terrain map from multiple sensor data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):278–292, Feb. 1992.

[14] C. Olson and L. Matthies. Maximum likelihood rover localization by matching range maps. In *International Conference on Robotics and Automation, Leuven (Belgium)*, pages 272–277, May 1998.

[15] P. Fillatreau and M. Devy. Localization of an autonomous mobile robot from 3d depth images using heterogeneous features. In *International Workshop on Intelligent Robots and Systems, Yokohama (Japan)*, 1993.

[16] M. Hebert. 3d landmark recognition from range images. In *IEEE International Conference on Computer Vision and Pattern Recognition, Champaign, Illinois (USA)*, pages 360–365, 1992.

[17] S. Betge-Brezetz, R. Chatila, and M.Devy. Object-based modelling and localization in natural environments. In *IEEE International Conference on Robotics and Automation, Nagoya, Japan*, 1995.