



A local method for contour matching and its parallel implementation

Samia Boukir, Patrick Bouthemy, François Chaumette, Didier Juvin

► To cite this version:

Samia Boukir, Patrick Bouthemy, François Chaumette, Didier Juvin. A local method for contour matching and its parallel implementation. Machine Vision and Applications, 1998, 10 (5-6), pp.321-330. <10.1007/s001380050082>. <hal-05729783>

HAL Id: hal-05729783

<https://hal.science/hal-05729783v1>

Submitted on 1 Sep 2026

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

A local method for contour matching and its parallel implementation

Samia Boukir¹, Patrick Bouthemy², François Chaumette², Didier Juvin³

¹ Laboratoire L3I, Université de La Rochelle, Avenue Marillac, F-17042 La Rochelle, France, e-mail: sboukir@gi.univ-lr.fr

² IRISA/INRIA, Campus Universitaire de Beaulieu, F-35042 Rennes, France

³ CEA-LETI / DEIN-SLA Saclay, F-91191 Gif sur Yvette Cedex, France

Received: 11 December 1996 / Accepted: 8 August 1997

Abstract. This paper presents a local approach for matching contour segments in an image sequence. This study has been primarily motivated by work concerned with the recovery of 3D structure using active vision. The method to recover the 3D structure of the scene requires to track in real-time contour segments in an image sequence. Here, we propose an original and robust approach that is ideally suited for this problem. It is also of more general interest and can be used in any context requiring matching of line boundaries over time. This method only involves local modeling and computation of moving edges dealing “virtually” with a contour segment primitive representation. Such an approach brings robustness to contour segmentation instability and to occlusion, and easiness for implementation. Parallelism has also been investigated using an SIMD-based real-time image-processing system. This method has been validated with experiments on several real-image sequences. Our results show quite satisfactory performance and the algorithm runs in a few milliseconds.

Key words: Contour matching – Directional convolution masks – Parallel processing – Real-time computer systems – Tracking of moving edges

1 Introduction

Matching features extracted from successive images is a common low-level issue in machine vision, and is considered as a key step in a large variety of tasks such as target tracking, depth recovery, object recognition, autonomous vehicle guidance, and many other tasks. This operation remains a difficult issue. Part of the difficulty comes from motion and depth discontinuities, and object occlusions that occur in real-world scenes. Furthermore, features can be matched in many ways.

Several algorithms for establishing correspondence have been proposed (Brown 1992), differing mainly on two elements:

- the level of description (primitives) at which the matching is attempted,
- the technique to match these primitives.

The choice of appropriate features is crucial to the correspondence computation. Low-level features, like pixels, involve a simple matching process, usually some form of area correlation (Goshtasby et al. 1984; Shen and Castan 1987; Brown 1992). However, these features are difficult to match correctly between successive frames because of their poor ambiguous descriptions. High-level features, such as surface patches, are easier to match, but hard to extract reliably from complex images (Chen and Nakatani 1992; Schweitzer 1993; Sabata and Aggarwal 1996). Therefore, the best compromise often appears to be the choice of edge elements, and, particularly, linear contour segments, as basic primitives (Medioni and Nevatia 1984; McIntosh and Mutch 1988; Deriche and Faugeras 1990; Lee and Yu 1990; Liu and Huang 1991; Palm et al. 1992; Li 1992; Crowley et al. 1992; Nasrabadi 1992; Tsai 1993; Wilson et al. 1994; Christmas et al. 1995). Description of such primitives consists of simple properties (length, orientation, ...) of the contour segments, and geometrical relations between them can also be considered. Matching is then usually achieved as follows. First, contour segments are extracted from the two considered images by detecting spatial edges, linking them, and determining a polygonal approximation. Then, correspondence between these two lists of segments is sought for, making use of the global parameters describing each contour segment. Ambiguities often appear which may be partly alleviated using contextual information. Thus, segments are matched constraining the potential matches of other segments so that geometrical relations are preserved.

In Medioni and Nevatia (1984), the correct match is obtained using a relaxation operation which tries to find the most consistent matches preserving the geometrical relations between features. Because of their ability to infer consistent interpretation from incomplete input, relaxation processes provide an attractive mean of inexact matching. However, they suffer from high computational and memory costs. The same matching scheme is also used in several other works (Li 1992; Nasrabadi 1992; Wilson et al. 1994 Christmas et al. 1995).

Another method that is widely used for matching linear features (and others) is based on the hypothesis prediction and verification process. The algorithm described in Ayache and Faugeras (1986) is a typical example of such a paradigm. In this approach, a hypothesis is first made to match two candidates for pairing. Then, this hypothesis is propagated over the other candidates, until it is rejected or it fits with most of them. Thus, another hypothesis is made and the process is repeated until all the possible hypotheses are tested, selecting the most adequate with respect to a quality measure. The major drawback of such an approach is also its computational costs. However, unlike the relaxation-based method, it has an intrinsically parallel nature since each hypothesis may be propagated independently.

Another way of facing this problem is to introduce a temporal recursive filter to track contour segments (Deriche and Faugeras 1990; Palm et al. 1992; Crowley et al. 1992; Zhang and Faugeras 1992) and to introduce a prediction step. Matching becomes a cyclic process composed of three main phases: predict, match, and update. Contour segments are represented as parametric primitives. These parameters are updated, usually using a Kalman filter. The correspondence between observed and predicted segments is often achieved through the use of similarity function based on the Mahalanobis distance between attributes of the segments. These methods are attractive because they limit the search area and handle ambiguities more effectively since they produce a prediction of the displacement from frame to frame.

Other typical methods for matching have also been reported in the literature, like dynamic programming methods (Maître and Wu 1988; Meer and Weiss 1992) and graph-matching methods (Horaud and Skordas 1989; Cinque et al. 1996; Gold and Rangarajan 1996).

It is often a poor image segmentation step that limits the effectiveness of these matching methods. The main difficulties stem from the loss of image entities due to under-segmentation and the presence of spurious entities due to noise or over-segmentation. Thus, the end-points and lengths of the corresponding contour segments may not necessarily be identical. Therefore, the matching technique should be able to handle inexact matching. The same problem occurs with the presence of occlusion. Furthermore, such methods may become too complex to be implemented for real-time performance.

Unlike previous methods, Hager and Toyama (1996) propose a real-time contour-segment-tracking algorithm which does not require dedicated processing boards, and avoids facing segmentation instability. Indeed, it does not aim at explicitly determining and using contour segment structures. A contour segment is characterized by its position and orientation, as well as its filter response. Given prior information, a window is acquired which, if the prior information is correct, leads to an edge which is vertical within the window. Convolving each row of the window with a scalar derivative-based kernel, summing down the columns of the window, and then finding the maximum value of this 1D response function, localizes the edges. This detection scheme requires orientation information to work correctly. If this information cannot be supplied from higher level geometric constraints, it is estimated at the image level. An expanded window corresponding to the predicted orientation is acquired, and the

summation step is performed along the columns and along two diagonal paths corresponding to two orientation offsets. In practice, the estimate of edge location will be biased. Additional accuracy can be gained by finding the zero-crossing of the second derivative at the computed orientation.

This method is highly time efficient, but works well only when the predicted orientation of the tracked segment is correct. Furthermore, the presence of several contour segments in the window of interest will greatly affect the tracking.

Our contour-segment-tracking method uses the following feature-level representation: a line segment (called a “virtual” segment here) is only represented by a list of pixels (its corresponding edge chain) and is defined by the two polar coordinate parameters (ρ, θ) of its support line. We use a different tracking procedure. Indeed, our method performs local matching of each pixel of the contour segment. Therefore, it combats noise and occlusion more effectively. A first version of this method has been presented in Boukir et al. (1993).

This paper is organized as follows. Section 2 briefly presents the main application that has motivated this work. Section 3 describes the first step of our tracking algorithm: the selection of a “virtual” segment in the first frame. Section 4 focuses on the tracking stage, which relies on an algorithm for detecting moving edges. Section 5 is devoted to the parallel implementation of our matching algorithm. Section 6 provides the results of experimental evaluations, followed by discussions and conclusions in Sect. 7.

2 Problem statement

Our matching technique was originally motivated by applications in the nuclear sector, but it is of general utility and can be applied to many other domains. More exactly, the aim is to assist a human operator in a maintenance task inside a power plant that is generally encumbered with numerous pipes. To this end, we have previously proposed (Chaumette et al. 1996) an active strategy for 3D structure estimation of cylindrical primitives adequately modeling pipes, using a mobile monocular camera. This method has been validated in real experiments and turned out to be a powerful. For each cylinder of interest, the 3D spatial structure estimation is accurately achieved if an optimal camera motion is selected. This optimal motion is then automatically computed by image-based closed-loop control laws (Espiau et al. 1992). Other primitives, like straight lines, spheres, etc., can also be considered. We control displacement and attitude of the camera in order to maintain the cylinder projection stationary in the centre of the image (gazing and fixation mechanisms), while the camera is rotating around the cylinder axis (structure from motion principle). We have called this estimation approach “structure from controlled motion”. This is an example of the active vision approach to scene analysis. This has the following implications in the way the matching problem is tackled:

- since displacement and attitude of the camera are controlled, we can use a prediction of the motion to ensure fast matching search;

- since the 3D recovery method only requires the computation of the parameters of the edge support line, we do not need to explicitly manipulate contour segments.

Such an approach requires that the image analysis step is performed in real-time. Therefore, we propose a real time solution for the tracking of a contour segment. We have examined the problem in two parts:

- select a contour segment from the first frame of the image sequence, and
- track the selected contour segment along the image sequence.

Besides, parallel processing has been taken into account to allow real-time implementation. Parallelism has been investigated using Sympati2 (Basille et al. 1988), an SIMD¹-based real-time image-processing system.

3 Selection of a contour segment

The problem we address in this section is to select a primitive, i.e., more specifically, an edge chain that will be tracked along the image sequence. The selection of the primitive is performed in three steps:

1. detecting spatial edges,
2. selecting a significant support line using the Hough transform (HT) technique, and
3. locating the longest edge chain lying on the selected support line.

3.1 Spatial edge detection

Several efficient edge detectors have been developed, showing comparable performance. We have chosen the Shen-Castan filter (Shen and Castan 1992), which is an optimal recursive filter whose transfer function has a discontinuity and a sign change at the origin. This filter offers good precision performance in localization, and is easily implemented on an SIMD architecture.

3.2 Selection of a support line

A support line is selected using the HT technique (Duda and Hart 1972) that was chosen for its robustness and its easiness to be implemented in an SIMD architecture. Having considered the polar coordinates (ρ, θ) as the line parameters, the Hough space will be the (ρ, θ) space, with $0 \leq \rho \leq \Delta/2$, where Δ is the image diagonal and $0 \leq \theta < 2\pi$. Then, finding the solution requires searching for the cell in the Hough space that receives the maximum number of votes in the accumulator array $H(\rho, \theta)$. Let us point out that such a technique allows us to easily make other choices (e.g., second highest pick) or to add supplementary constraints (e.g., highest pick within a given orientation interval) for selecting the contour segment of interest. Only the edge points take part in the voting procedure. In order to reduce the processing

time, we proceed in two steps: a first step with a coarse sampling for ρ and θ giving rise to a maximum $(\rho_{max}, \theta_{max})$ which corresponds to the densest region within the parameter space. Next, the Hough space is reduced to a window limited round the values $(\rho_{max}, \theta_{max})$, and a higher sampling for ρ and θ is used. This second step provides the parameters $(\tilde{\rho}, \tilde{\theta})$ related to the selected support line on which we proceed to detect the longest edge chain.

3.3 Edge chain location

The chaining algorithm that we have developed is simple. Indeed, the previous stage provides an information of key interest, namely, the parameters $(\tilde{\rho}, \tilde{\theta})$ related to the selected support line of the points to be chained. The chaining is performed according to the following steps:

1. edge chain computation: edge chain points are looked for in an area limited to a strip centered around the support line defined by the parameters $(\tilde{\rho}, \tilde{\theta})$,
2. small edge chains are discarded and neighboring ones are merged,
3. selection of the longest edge chain: it represents the contour segment that is to be tracked along the image sequence. Indeed, as explained later, we do not need to perform a polygonal approximation. This level of representation is sufficient for our matching technique.

Let us note that only the first image of the sequence is concerned with this preliminary stage. The computational constraint necessary to ensure real-time contour-segment-tracking mainly concerns the matching stage (next stage).

4 Matching contour segments

The technique for matching contour segments that we have developed exploits a method for determining local moving edges (MEs) previously designed in Bouthemy (1989). Let us first outline the main principles of the latter, then we will describe the extensions we have introduced.

4.1 Local matching of edge points

The edge-point-matching procedure relies on the modeling of a moving edge as a spatio-temporal surface patch in the image sequence considered as a volume (Bouthemy 1989). A likelihood ratio test allows us to determine its location, its local spatial orientation, and its displacement component orthogonal to the edge $V^\perp$. An image sequence is considered as a 3D volume with two spatial dimensions (x, y) and a temporal dimension (t) . In this volume, an ME locally generates a small surface patch. This surface may be locally approximated by a planar patch defined by the following parameters (Fig. 1):

1. position (x_0, y_0) and angle θ with respect to the x -axis,
2. angle ψ with respect to the t -axis, whose tangent provides the normal displacement $V^\perp$ of the ME in the image plane ($V^\perp = \tan \psi$).

¹ Single Instruction Multiple Data

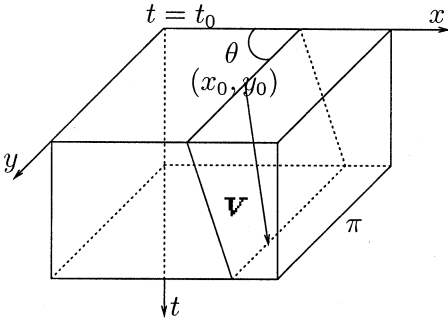


Fig. 1. Local modeling of a moving edge as a spatio-temporal surface patch (here, a planar patch)

The planar patch model is the most simple, and is easily tractable. More complex surface patch models can also be used, provided they may be parametrized (Marcé and Bouthemy 1987). Let us consider an elementary volume π located at (x_0, y_0, t_0) in the 3D volume (x, y, t) . The method described in Bouthemy (1989) supplies the detection of such surface patches and the estimation of their parameters θ and $V^\perp$. This method is based on a log-likelihood ratio test taking into account hypotheses of local configurations. These hypotheses stemming from the spatio-temporal modeling of the ME are associated with two competing situations: either there is no ME inside π , or there is one. Each hypothesis is associated with a likelihood function, which is the joint probability density function of the intensity variables within the elementary volume π , which are assumed to be independent Gaussian variables. It turns out that the computation of this test leads to a simple convolution operation with a set of masks associated with pre-defined geometric configurations of the ME. In this primary version, all allowed quantized values of both orientation and displacement are considered in a systematic way.

Our choice of such an approach has been mainly motivated by the following factors:

- the attractive property of deriving an elaborate decision criterion with a convolution-like efficiency, which is a quite valuable feature to achieve a real-time implementation.
- the successful handling of various kinds of edges (in particular, those corresponding to occlusion boundaries) (Bouthemy 1989),
- the absence of restrictions concerning the extent of measurable motion magnitude.

However, the determination of the whole set of MEs within an image is time consuming, especially when the density of spatial edges is high. On the other hand, as stated before, our reconstruction strategy tackles only one contour segment at a time so as to optimally obtain the 3D spatial structure. Therefore, the ME computation concerns only a restricted subset of the spatial edges within the image. The processing time is then significantly reduced, as demonstrated later.

4.2 Matching “virtual” segments

The aim here is to track the selected “virtual” segment along the image sequence. Let us emphasize that our track-

ing method does not require spatial edge detection for each frame of the sequence. This processing, as previously explained, concerns only the first frame on which we select a contour segment to track. Thus, our method is not a classical contour-based matching method with edge detection and contour segment determination steps at each frame of the sequence. We propose an original method arising from the ME computation algorithm. This pixel-based method relies on three basic steps, explained in the following subsections.

4.2.1 ME computation

Given a pixel list L_i corresponding to the current “virtual” segment S_i in image i of the sequence, we want to determine their corresponding points in image $i+1$, using a new version of the ME computation algorithm adapted to our matching problem. The ME determination is now achieved in the following steps.

a. Optimal convolution mask: The direct computation of $\hat{\theta}$, the optimal estimate of the spatial orientation of S_i is available here, as explained below in Subsect. 4.2.2. This allows the derivation of perfectly suitable convolution masks improving the robustness of the log-likelihood ratio test. In the primary version (Bouthemy 1989), a coarse quantization is used (for instance, every 45° for angle θ) and all possible values of θ need to be evaluated in turn. On the contrary, in the present case, a much finer accuracy (1°) can be considered, since only one θ value is relevant. This precision greatly improves the reliability on the measure of $V^\perp$, the normal displacement. This represents a significant optimization on the computational load. Now, the ME determination only involves the estimation of $V^\perp$, since the location and the spatial orientation θ of MEs have been previously and “externally” determined.

Let us denote by $M_{|\theta^*}^1$ and $M_{|\theta^*}^2$, the convolution masks associated with θ^* , the integer value closest to $\hat{\theta}$. Images i and $i+1$ are convolved with $M_{|\theta^*}^1$ and $M_{|\theta^*}^2$. These masks are the same and can be pre-computed when the displacement magnitudes are limited to integer values in the pixel grid.

b. Search area: Due to the aperture problem, we can locally estimate only one component of the edge displacement, i.e., the component normal to the edge, leading to a 1D search area in that direction as illustrated in Fig. 2, where the introduced notations have the following meaning:

- $P_k \in L_i$: edge point of edge segment S_i .
- Q_k^j : possible new location of point P_k in the direction normal to the edge at P_k , in the image $i+1$, $j \in [-J, J]$ where J is the maximal allowed displacement magnitude.

According to the direction of the normal at P_k , the considered displacements do not necessarily correspond to entire locations Q_k^j within the image grid. On the other hand, dealing with non-integer values of displacement requires either a computation of new convolution masks or some interpolation of the intensity function. Both involve significant additional computational costs. This problem can be alleviated by considering only four main search directions δ :

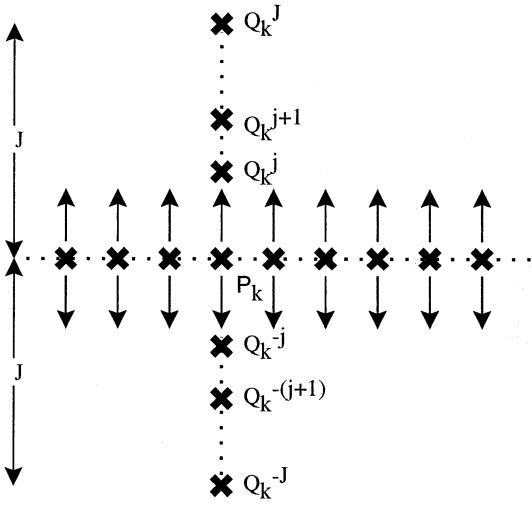


Fig. 2. 1D search area for the ME determination

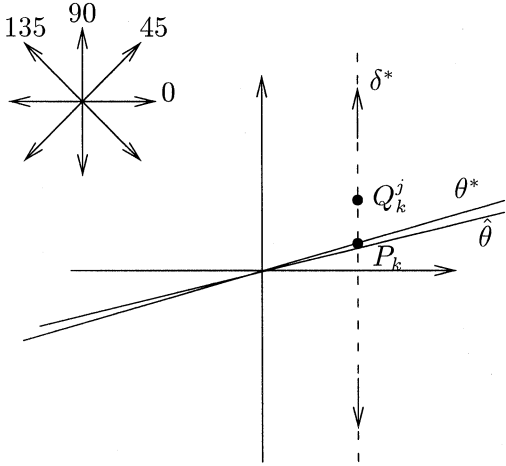


Fig. 3. Selection of the search direction

$\delta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$, inducing integer locations Q_k^j after displacement (Fig. 3). Among these directions, we select the direction δ^* closest to the normal to the direction θ^* .

This choice seems to be a suitable compromise if real-time implementation is aimed for. Furthermore, let us recall that our final objective is the global matching of two successive positions of a contour segment. Hence, we are not interested in the 2D displacement of each ME. We need to track this segment to update the values of the parameters (ρ, θ) related to the contour segment at the next instant.

The correspondent of each P_k of the list L_i are looked for within a pre-determined search interval in the direction δ^* . If the next location of the current segment can be predicted, then the search area can be limited to be around this prediction. In our case, we take advantage of the closed-loop approach used in our 3D reconstruction method to optimally adapt the 1D-search window (values of the window middle point and the window size).

c. ME determination: Let us consider a pixel P_k belonging to the list L_i and a subset of positions $Q_k^j \in [-J, J]$ lying in the direction δ^* in the next image. The corresponding log-likelihood ratio values ζ_j are given by the absolute value

of the sum of the convolution values computed at P_k and Q_k^j , using the masks $M_{|\theta^*}^1$ and $M_{|\theta^*}^2$, respectively for $j \in [-J, J]$. If the maximum value for ζ_j is greater than some predetermined threshold λ , then an ME is claimed to be present and the corresponding point Q_k^{j*} is stored in the pixel list L_{i+1} . Thus, a new list of pixels is obtained, whose points are the estimated correspondents of those in the list L_i .

4.2.2 Estimation of the ME support line parameters

Given the ME list L_{i+1} obtained from the previous step, a least squares technique is used to estimate the parameters $(\hat{\rho}_{i+1}, \hat{\theta}_{i+1})$ related to the support line in image $i+1$. Then, the least squares residuals are computed, and are considered as a measure of the quality of the estimation:

$$\varepsilon_{quad} = \frac{1}{n} \sum_{j^* \in L_{i+1}} (x_{j^*} \cos \hat{\theta}_{i+1} + y_{j^*} \sin \hat{\theta}_{i+1} - \hat{\rho}_{i+1})^2, \quad (1)$$

where n denotes the number of points in the list L_{i+1} .

If ε_{quad} is greater than a pre-determined maximum value, we conclude that a mismatch has occurred. Indeed, a global ambiguity may arise in an image that cannot be settled by the ME computation algorithm: two contours having the same intensity contrast and the same sign contrast may be present in the search area, and no definite way exists to select the correct one. Of course, this extreme case seldom occurs. Furthermore, the algorithm may locally detect some false MEs. These outliers are rejected using the following test:

$$|x_{j^*} \cos \hat{\theta}_{i+1} + y_{j^*} \sin \hat{\theta}_{i+1} - \hat{\rho}_{i+1}| < \varepsilon_{out}, \quad (2)$$

where ε_{out} is a pre-determined threshold. The list L_{i+1} is then updated and the least squares is performed again to refine the estimation of the current segment parameters. Let us point out that a “maintenance” of the list L_{i+1} is necessary in order to ensure a reliable tracking of the “virtual” segment S_{i+1} . Indeed, some points located at the extremities of the segment might be unmatched by looking for a correspondent in the normal direction due to the sliding movement of the segment along itself. Furthermore, partial occlusion may occur and the list L_{i+1} must be updated once the occluded part of the segment has appeared again. The missing points of the list L_{i+1} are easily recovered using a chaining procedure around the support line related to L_{i+1} . Finally, the few spurious points Q_k^{j*} , if any, must be replaced with the correct ones situated on the support line.

4.2.3 Discussion

Our method has the following advantages:

- only one data structure is considered: the array of pixels;
- it is not affected by contour segmentation instability and it is robust to occlusion. Indeed, we do not explicitly use contour segment structures but edge chains which do not require a polygonal approximation step. The matching of the edge chains only requires a sufficient number of local matching of edge points to be correctly performed, thus, ensuring robustness to partial occlusion;

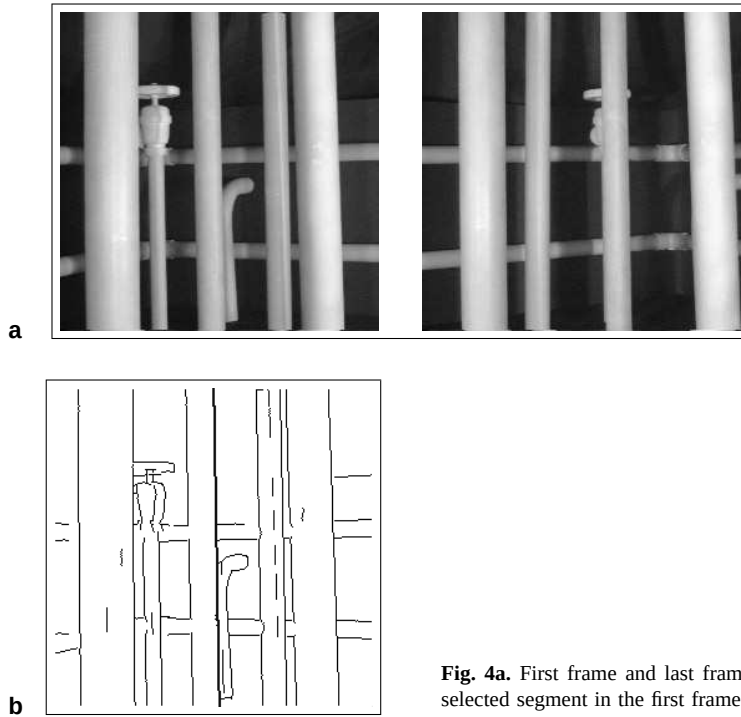


Fig. 4a. First frame and last frame of “pipes” sequence; image size: 256×256 . **b** Support line of the selected segment in the first frame overdrawn in black on the edge image (in the middle of the image)

- it is easy to implement on image-processing boards due to its point-wise local properties. This is an advantage over matching approaches that use higher level primitives;
- the extracted edge chain can be sub-sampled to speed up the tracking procedure.

5 Parallel implementation

Matching algorithms is usually characterized by an algorithmic complexity of high order and by the large quantity of processed data. Acceptable matching computational time for industrial applications would not be possible without real-time solution. The use of specialized circuits for image processing (Hussain 1991) may be a possibly efficient solution, but this solution is costly. On the other hand, there exist high-performance parallel boards, supporting several image-processing operations, like, for instance, the Sympati2 SIMD system (Basille et al. 1988). This machine is well suited to run image-processing algorithms. It has already been evaluated on industrial applications such as inspection and quality control.

Our matching method provides a very simple mechanism of contour segment pairing. This simplicity is essential when considering parallel implementation. Furthermore, it has an intrinsically parallel nature, and maps well to the SIMD model of processing thanks to its local properties.

In the following, we will describe the parallel implementation of our matching method which includes, as previously explained, two parts: the selection stage which consists of selecting a contour segment in the first image, and the tracking stage of the selected contour segment over successive frames of the sequence. Both modules have been implemented on the real-time image-processing system Sympati2 (see appendix A for a brief description of this machine).

5.1 Parallel implementation of the selection stage

The time complexity of this stage obviously depends on the edge density in the first frame of the image sequence. It also depends on both the number of processing elements involved in the processor array structure, and the number of considered straight line directions in the Hough space. For instance, with a configuration of 32 PE, Sympati2 spends around 225 ms to achieve the HT (well known for its computational costs) on a 256×256 image with 5% of edge points, and when considering all the possible straight line directions in the discrete Hough space (Dorn and Adam 1989). For the same configuration, Sympati2 performs the spatial edge detection (Shen-Castan filter) in about 35 ms and the chaining in less than 8 ms. The whole selection stage then takes around 268 ms on Sympati2. Let us point out that the relative slowness of this step does not compromise the performance of our active 3D reconstruction approach (Chaumette et al. 1996), since it is performed only once. However, the tracking step must be compatible with the sampling rate of the closed-loop control law which is at least 10 Hz (100 ms per image).

5.2 Parallel implementation of the tracking stage

The performance obtained for the tracking stage is very attractive in Sympati2. Indeed, it takes around 30 ms for a 256×256 image in case of a 32 PE configuration. This processing time depends on both the orientation and the length of the tracked contour segment, and also the allowed maximal normal displacement magnitude. The search interval, which has been primarily set to ± 7 pixels in all our experiments, can be reduced to ± 2 after the convergence of the control law. This has a consequence of reducing the computation time of 50%. Let us note that this implementation

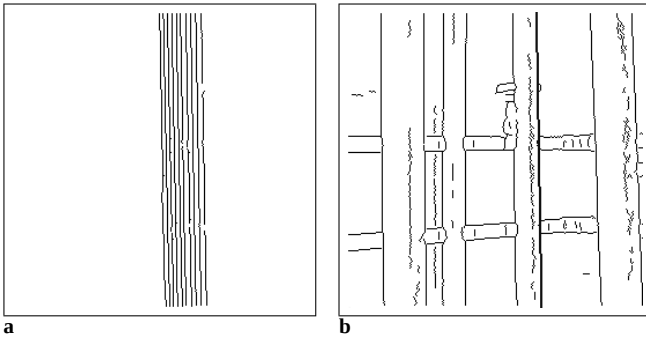


Fig. 5a,b. “Pipes” sequence: **a** Set of the successive estimated segment positions in the image sequence corresponding to the selected segment. **b** Location of the support-line (overdrawn in black) of the correctly tracked segment in the last image of the sequence

can also permit the simultaneous tracking of several contour segments with similar orientations (at least five with the present parallel version of the tracking algorithm) in less than 100 ms. We recall that this figure represents the sampling rate of the closed-loop control law used in our real-time 3D recovery system (Chaumette et al. 1996).

6 Experimental results

The proposed algorithm for tracking contour segments has been applied successfully to various real images. Only the results of two experiments will be described in this section. We use a convolution mask of size 3×3 , a maximal search interval of ± 7 pixels, and a likelihood ratio threshold $\lambda = 1500$ for all experiments.

“Pipes” scene example: Figure 4a shows both the first and the last frame of an image sequence acquired by a moving camera in an area containing several pipes. This environment illustrates the interior of a nuclear power plant. Figure 4b contains the contour segment that has initially been selected (note the thick vertical line). The results related to the tracking of this segment along the sequence are displayed in Fig. 5. On the left, the resulting sequence of the successive estimated positions (for all the frames) of the tracked contour segment, and, on the right, the location of the support line of the tracked segment in the last image of the sequence which perfectly fits the corresponding edge line. These results illustrate the reliability of our matching technique. Let us recall that our tracking method does not require spatial edge detection in each frame of the sequence. This processing step is needed only once at the beginning of the tracking procedure to select a contour segment. Thus, our method is not a classical contour-based matching method with explicit edge detection at each frame of the sequence. This shows once again the simplicity and the great real-time capabilities of our matching method. To check the reliability of our tracking process, we have nevertheless extracted spatial edges in the last frame of the sequence and overprinted the tracked edge chain on the resulting contour image at the same time instant as shown in Fig. 5b, in order to demonstrate the performance of our method. Let us point out that a global geometric matching approach would fail to distinguish between the left and the right limbs of the cylinder

(note that the other contour segments of the scene are also prone to bring matching ambiguities). Such a shortcoming is not encountered by our algorithm, which licitly uses the intensity contrast as local property for MEs. Moreover, since the camera is rotating around the scene, the projection of the background around the segment of interest is changing, and this does not affect our matching algorithm. The proposed method needs only one parameter to be set: λ . It does not require a fine-tuning; the same value (1500) has been successfully used for all our experiments.

Parallel implementation: The selection stage takes 303 ms on Sympati2 with a 32 PEs configuration: 40 ms for the spatial edge detection (Shen-Castan filter), 10 ms for the chaining, and 253 ms for the HT. For the same configuration of Sympati2, the current computation of the matching step takes 32 ms, which is a very satisfactory performance. Note that, for this experiment, the vertical orientation of the segment in this example is an advantage for the computational performance, whereas the almost maximal length of the segment is a disadvantage.

“Calibration board” scene example: Figure 6a shows both the first and the last frame of an image sequence of a scene containing a vertical cylinder and a calibration set-up. The motion of the camera that has been executed between the first and the last image of the sequence consists of a translation of 500 mm in the horizontal plane and a rotation of 35° around the vertical axis of the camera (the objects are located at about 1 m from the camera). Figure 6b shows the contour segment that has been selected by the HT performed on the right part of the first frame.

Once more, our algorithm ensures a perfect tracking of the selected segment despite the complex motion of the camera. Figure 7, which is structured in the same way as Fig. 5, contains the obtained results. Let us point out that a global approach would have some difficulty to overcome the matching ambiguity that arises in this experiment. Indeed, there are two very close candidate segments for matching, and both have the same orientation and length. The results of this second example show again the robustness of our approach to such ambiguities. In our method, the whole matching of two segments only requires a sufficient number of local matching of edge points to be correctly achieved. Therefore, this is a robust technique able to deal with local ambiguities and partial occlusions.

Parallel implementation: The selection stage here takes 232 ms on Sympati2 with a 32 PEs configuration: 40 ms for the spatial edge detection (Shen-Castan filter), 8 ms for the chaining, and 184 ms for the HT. For the same configuration of Sympati2, the current computation of the matching step takes 28 ms.

Application of the tracking method to a visual servoing task: The main application that has motivated our work on matching techniques is a visual servoing task. More exactly, we aim at optimally recovering the 3D structure of a scene using an image-based closed-loop control law. This visual servoing approach requires a real-time tracking of image primitives.

In the experimental set-up of IRISA, image-processing part is performed on a conventional image-processing board

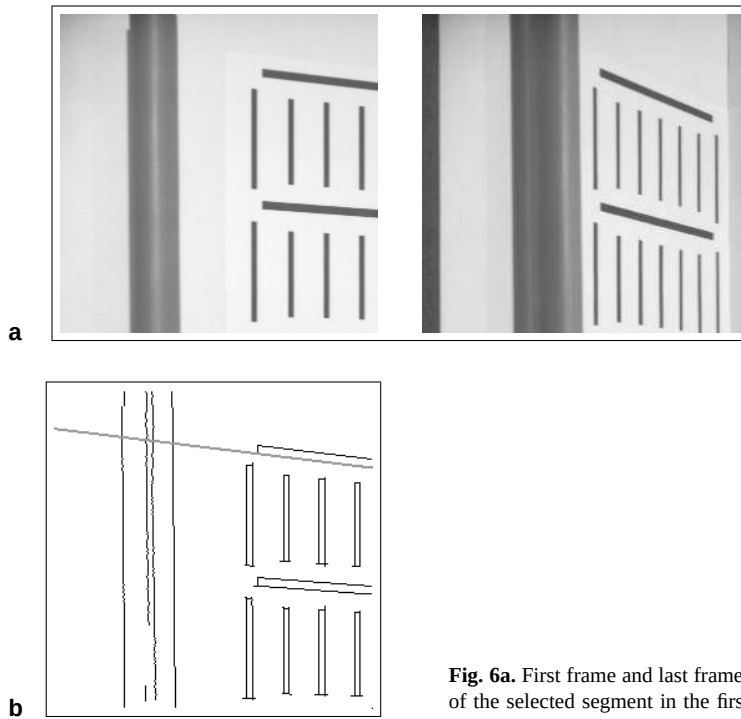


Fig. 6a. First frame and last frame of “calibration board” sequence; image size: 256×256 . **b** Support-line of the selected segment in the first frame overdrawn on the top of the edge image

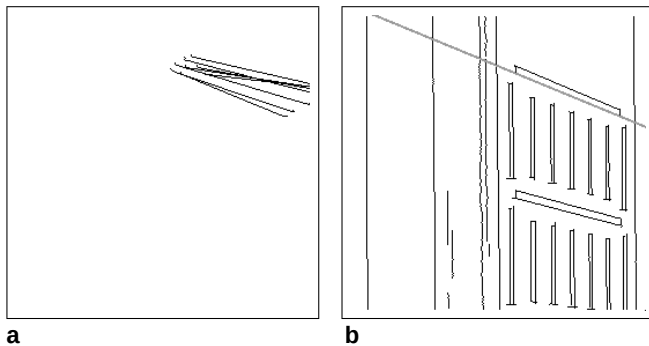


Fig. 7a,b. “Calibration board” sequence: **a** Sequence of the successive estimated segment positions corresponding to the selected segment. **b** Location of the support-line (overdrawn in black) of the correctly tracked segment in the last image of the sequence

EDIXIA-1000. Therefore, to validate our real-time 3D structure recovery technique on this system, our contour-segment-tracking method has also been implemented on EDIXIA-1000.

EDIXIA-1000 consists of a standard VME board ensuring the acquisition of images of size 512×730 pixels $\times 8$ bits and offering real-time hardware operators like filtering, convolution, etc. The main processor in this system is a 68020-based CPU board.

The EDIXIA-1000 language is C-ANSI, with possible system calls to a library including real-time hardware image-processing operators.

Our tracking algorithm is implemented on EDIXIA-1000 as follows:

- convolution operations are performed by the available real-time convolution operator;

- the remainder of the algorithm is processed by the 68020-based CPU board. To enable real-time implementation on this mono-processor board, the points of the extracted edge chain are sub-sampled (a maximum of 40 points).

The performance achieved is about 1 s for the selection stage, and between 30 ms and 40 ms for the tracking step. Let us note that the simultaneous tracking of three contour segments with similar orientations has also been implemented on this processing board, and the obtained performances remain very attractive (between 60 ms and 80 ms). Numerous real-time 3D structure recovery experiments have been successfully realized on the experimental set-up of IRISA (Boukir 1993; Chaumette et al. 1996; Marchand et al. 1996). The reconstructed scenes include partly occluded cylinders and polygonal objects which have been properly tracked over a long time interval. This shows once again the efficiency of our tracking algorithm and the capability of real-time implementation on conventional machines, too.

7 Conclusion

We have described a local approach for matching contour segments in an image sequence. Our method was originally motivated by a 3D structure recovery task exploiting an active vision approach based on a closed-loop scheme (visual servoing). Such an approach requires a real-time image-processing step. Therefore, we propose an original and robust method that perfectly satisfies both robustness to complex situations and processing time constraints imposed by the application. This method is based on a spatio-temporal modeling of an ME. A likelihood ratio test enables us to achieve its detection and the estimation of its related attributes. The computation of this test leads to a simple convolution operation. The method described in this paper han-

dles matching of contour segments without explicitly building and manipulating such data structures. This attractive property allows us to easily implement our algorithm on a real-time image-processing system. The current computation of the matching modul takes about 30 ms per image on the image-processing board Sympati2. Our matching method has been validated with several experiments on real image sequences. This approach takes also advantage of the way we have coped with the 3D structure recovery problem. Finally, let us emphasize that attractive features of our method (compared to other global matching approaches) are its generality and its versatility. Indeed, it can easily be extended to use other features such as elliptical arcs or circle segments.

Appendix A Sympati2 architecture

The Sympati2 system (Basille et al. 1988) is a linear processor array structure with 2D addressing capabilities. The main feature of this linear structure is the helicoidal scheme adopted for the data organization, which makes it possible to access, either on a row or on a column, any M pixel segment, where M is the number of processing elements involved scalable from 32 to 256.

The processing elements structure consists of two main parts: the processing part, concerned with the operations on the pixels values, and the addressing part, allowing each PE to calculate the required pixel address in its own memory bank. More details are given below.

- The processing part consists of a 16-bit ALU for usual operations (binary, arithmetics and shifting operations), a scratch pad of registers, a regular interconnection system, enabling each processor to have access, in one machine cycle, to data located in processors up to distance 3, and four flags necessary for performing conditional sequences.
- The addressing part makes it possible to address the memory either in helicoidal mode or in tabulation mode. In helicoidal mode, it evaluates the considered pixel coordinates (i, j) from the current segment coordinates sent by the command unit, and then performs the helicoidal transform: $@ = i.T/M + j/M$, which is the address of pixel (i, j) in the memory bank of PE N ($N = (i + j) \text{Mod. } M$) for an $T \times T$ image and a configuration of M PE.

Sympati2 provides row-by-row or column-by-column scanning possibilities while keeping neighborhood relationships. Thus, it is well adapted to point operations and local neighborhood operations (Basille et al. 1992).

The Sympati2 language is 4LP (Low-Level Language for Line Processor). This language dedicated to an SIMD architecture has an attractive characteristic: the parallelism is partly hidden for the programmer. Indeed, in this parallel language, processing is simply expressed for a point of the image and its neighbors. The same processing is therefore automatically propagated for other concerned pixels in the whole image.

Acknowledgements. This work has been partly supported by CEA under contract SAV16849/VCH, by the French PRC Programme “Man-Machine Communication” under contract MRE91S0269, and by the Brittany County Council (“Région Bretagne”) for contribution to student grant. We would like to thank M. Viala and J.P. Guérin for the parallel version of the Hough technique on Sympati2, and Michel Ollivier from EDIXIA for the implementation on the EDIXIA image-processing board.

References

1. Ayache N, Faugeras OD.(1986) Hyper: a new approach for the recognition and positioning of two-dimensional objects. *IEEE Trans PAMI* 8(1):44–54
2. Basille JL, Essafi H, Juvin D, Latil JY (1988) SYMPATI2: A 1.5D processor array for image application. *Proc. of 4th Conf. EUSIPCO*, Grenoble, France
3. Basille JL, Fernandez P, Komen ER (1992) Recursive neighbourhood operations on the linear processor array SYMPATI-2. *Image Vision Comp* 10(9):625–630
4. Boukir S (1993) Reconstruction 3D d’un environnement statique par vision active. PhD thesis, University of Rennes 1, France
5. Boukir S, Bouthemy P, Chaumette F, Juvin D (1993) Real-time contour matching over time in an active vision context. *Proc. of 8th SCIA, Scandinavian Conference on Image Processing*, Tromsø, Norway, pp 113–120
6. Bouthemy P (1989) A Maximum Likelihood Framework for Determining Moving Edges. *IEEE Trans PAMI* 11(5):499–511
7. Brown LG (1992) A survey of image registration techniques. *ACM Comput Surv* 24(4):325–376
8. Chaumette F, Boukir S, Bouthemy P, Juvin D (1996) Structure from controlled motion. *IEEE Trans PAMI* 18(5):92–504
9. Chen YL, Nakatani H (1992) Image region correspondence by structural similarity. *Proc. of 11th ICPR, International Conference on Pattern Recognition*, The Hague, The Netherlands, 3:18–21
10. Christmas WJ, Kittler J, Petrou M (1995) Structural matching in computer vision using probabilistic relaxation. *IEEE Trans PAMI* 17(8):749–764
11. Cinque L, Yasuda D, Shapiro LG, Tanimotos S, Allen B (1996) An improved algorithm for relational distance graph matching. *Pattern Recognition* 29(2):349–359
12. Crowley JL, Stelmazyk P, Puget P (1992) Measurement and integration of 3D structures by tracking edge lines. *Int J Comput Vision* 8(1):29–52
13. Deriche R, Faugeras O (1990) Tracking line segments. *Image Vision Comput* J 8(4):261–270
14. Dorn F, Adam P (1989) Evaluation et performance de la structure Sympati2. Adaptation de la Transformée de Hough. Technical Report, CEA Saclay, France, SIR89.32
15. Duda RO, Hart PE (1972) Use of the Hough transformation to detect lines and curves in pictures. *Commun ACM* 15 (1):11–15
16. Espiau B, Chaumette F, Rives P (1992) A new approach to visual servoing in robotics. *IEEE Trans Robotics Automation* 8(3):313–326
17. Gold S, Rangarajan A (1996) A graduated assignment algorithm for graph matching. *IEEE Trans PAMI* 18(4):377–388
18. Goshtasby A, Gage SH, Bartholic JF (1984) A two-stage cross correlation approach to template matching. *IEEE Trans PAMI* 6:374–378
19. Hager GD, Toyama K (1996) X Vision: combining image warping and geometric constraints for fast visual tracking. *Proc. of ECCV’96*, Cambridge, U.K, pp 507–517
20. Horaud R, Skordas T (1989) Stereo correspondence through feature grouping and maximal cliques. *IEEE Trans PAMI* 11:1168–1180
21. Hussain Z (1991) Digital image processing. Practical applications of parallel processing techniques. Ellis Horwood Press, England
22. Lee HJ, Yu DJ (1990) Line-Based Structural Matching Via Segment Splitting. *Pattern Recognition Lett* 11:181–189
23. Li SZ (1992) Matching: invariant to translations, rotations and scale changes. *Pattern Recognition* 25(6):583–594
24. Liu Y, Huang TS (1991) Determining straight line correspondences from intensity images. *Pattern Recognition* 24:489–501

25. Maître H, Wu Y (1988) A new dynamic programming method for stereovision ignoring epipolar geometry. Proc. of the 9th ICPR, International Conference on Pattern Recognition Rome, Italy, 1:146-148
26. Marcé L, Bouthemy P (1987) Determination of a depth map from an image sequence. Proc. of the 3rd Int. Conf. on Advanced Robotics ICAR, Paris, France, pp 221-232
27. Marchand E, Chaumette F (1996) Controlled camera motions for scene reconstruction and exploration. Proc. of CVPR'96, San Francisco, USA, pp 169-176
28. McIntosh JH, Mutch KM (1988) Matching Straight Lines. Comput Vision Graphics Image Process 43:386-408
29. Medioni G, Nevatia R (1984) Matching images using linear features. IEEE Trans PAMI 6(6):675-685
30. Meer P, Weiss I (1992) Point/line correspondence under 2D projective transformation. Proc. of ICPR'92, International Conference on Pattern Recognition, The Hague, The Netherlands, 1:399-402
31. Nasrabadi NM (1992) A stereo vision technique using curve-segments and relaxation matching. IEEE Trans PAMI 14(5):566-572
32. Palm HC, Osteveid E, Grinaker S (1992) Template matching used in time contextual segmentation. Proc. of ICPR'92, International Conference on Pattern Recognition, The Hague, The Netherlands, 3:205-208
33. Sabata B, Aggarwal JK (1996) Surface correspondence and motion computation from a pair of range images. Comput Vision Image Understanding 36(2):221-232
34. Schweitzer H (1993) A surface matching algorithm from two perspective views. Proc. of CVPR'93, Conf on Comput Vision Pattern Recognition pp 742-743
35. Shen J, Castan S (1987) Fast binary image correlation algorithm for stereovision. Proc. on 5th SCIA, Stockholm, 1:317-324
36. Shen J, Castan S (1992) An optimal linear operator for step edge detection. Comput Vision Graphics Image Process (Graphical Models and Image Processing) 54(2):13-17
37. Tsai FCD (1993) Robust Affine Invariant Matching With Application To Line Features. Proc. of CVPR'93, Conf. on Computer Vision and Pattern Recognition, pp 393-399
38. Wilson RC, Evans AN, Hancock ER (1994) Relational matching by discrete relaxation. British Machine Vision Conference, York, England, 1:43-53
39. Zhang Z, Faugeras O (1992) 3D Dynamic Scene Analysis - A Stereo Based Approach. Springer, Berlin Heidelberg New York



Samia Boukir received the computer science engineering degree from the University of Algiers, Algeria, in 1989, and the Ph.D. degree from the University of Rennes, France, in 1993. Since 1993, she has been an assistant professor at the University of La Rochelle, France, and has worked at L3i (Laboratoire d'Informatique et d'Imagerie Industrielles) on image processing. Her current research interests include image processing and computer vision, particularly in the area of three-dimensional and dynamic scene analysis.



Patrick Bouthemy graduated from Ecole Nationale Supérieure des Télécommunications, Paris, in 1980, and received the Ph.D degree in Computer Science from the University of Rennes, France, in 1982. From December 1982 until February 1984, he was employed by INRS-Télécommunications, Montréal, P.Q., Canada, in the Department of Visual Communications. Since April 1984, he has been with Inria, at Irisa in Rennes. He is currently "Directeur de Recherche" Inria. His major research interests concern both visual perception and image-based communication: motion analysis (rigid motion and deformable motion), statistical models for image sequence processing (MRF models, robust estimation, Bayesian estimation), dynamic scene interpretation, active vision, and content-based video indexing.



François Chaumette graduated from Ecole Nationale Supérieure de Mécanique, Nantes, France in 1987 and received the Ph.D degree in Computer Science from the University of Rennes, France, in 1990. In 1991, he received the AFCET / CNRS Prize for the best french thesis in Automatic Control. Since 1990, he has been with INRIA, at IRISA in Rennes, France, where his current title is "Chargé de Recherche". His research interests include robotics, computer vision, and especially the coupling of these two research domains (vision-based control, active vision and purposive vision).



Didier Juvin is graduated Engineer in Electronics from The Institut Supérieur d'Electronique de Paris since 1977. He first worked at THOMSON on simulators developing image synthesis algorithms. Then he joined CEA (French Atomic Energy Commission) in 1980 at the DEIN (Electronics and Nuclear Instrumentation Department) and wrote his Ph.D thesis in image processing in 1983 on overlapping part 2D recognition, transferred to GIXI (VX image processing family). He then developed with IRIT a new parallel concept for image processing. He was responsible of the SYMPATI 2 project that led to an industrial product (CENTRALP: OPENVISION Systems). He is Head of the Software and Architecture Laboratory, a team of 53 engineers and 12 thesis inside the LETI. His principal work is concerned with developing parallel computer architectures devoted to image processing and pattern recognition.