MVA'90 IAPR Workshop on Machine Vision Applications Nov. 28-30,1990, Tokyo

A new method for recognizing and locating objects by searching longest paths.

G.BERTRAND and F.BOUSSOFIANE

Laboratoire IAAI. Groupe ESIEE.
2 Bd Blaise Pascal BP 99
93162 Noisy-le-Grand,France.

Abstract

This paper presents a new method for recognizing and locating an object in an image. The contours of the shapes are described by line segments. The basic idea is to generate diagonal paths in a table where the columns represent the model segments and the rows the scene segments. Each path represents a matching between a part of the model and a part of the scene contour. The longest path is considered as the best match.

1 Introduction

In robotic applications and industrial vision one of the most important problem is to identify and locate objects in a scene.

When objects are isolated, classical methods are based on a description of shapes by global features such as area, size and moments. These features are reliable but become totaly unusable in the case of objects deformations or overlapping objects.

Hence the use of local features which describe limited portions of the shape is necessary for partially altered or occluded parts.

One approach is to describe each object by a relational graph. Each node in the graph represents a feature and an arc between two nodes expresses the relationship or compatibility between the features. The matching process consists of finding a sub-graph isomorphism between the image and the model graphs. This isomorphism may be achieved by a relaxation technique, a tree search technique or a maximal clique finding method [BHA 84] [BAL 82] [SKO 88]. The Local Features Focus (L.F.F) method developped by Bolles and Cain [BOL 82] identifies objects partially visible by finding maximal cliques to establish a first matching hypothesis.Then a technique of prediction and verification is applied. The first limitation of these methods is the combinatorial explosion when the number of features increases. The approach developped by Ayache [AYA 86] is based upon matching simple descriptions of the scene and the model by a technique called HYPER (HYpothesis Predicted and Evaluated Recursively) of hypotheses generation and verification coupled with a recursive estimation of the model to scene transformation. Also based upon the principle of prediction-verification, the method developped by Bouyakhf [BOU] takes into account the structural relationship between the segments to initialize and propagate the matching hypotheses. These prediction and verification techniques are fast, accurate and robust to noise. McIntoch and Mutch [McIN 88] present a new approach to match line segments extracted from image pairs. A match function determines the strength of the similarity between a pair of lines.Pairs which are mutually best matched are considered as corresponding lines. The results indicate that the method works well in motion and stereo cases. Another approach is that of Lee and Yu [LEE 89]. Their method relies on line-based structural matching. To measure the similarity between line segments, a set of neighboring segments is found for each segment. In order to get accurate and reliable measures, each segment is split into short parts of the same length. Relationship measures are computed between each segment and its neighbors. A similarity measure permits to find the segment-neighbor pairs which correspond in the two images. Dynamic programming recently appeared as a tool for matching images under severe distorsions and noise conditions. Maitre and Wu propose an elastic matching method based on dynamic programming to solve the problem of elastic distortions [MAI 88]. They have also proposed other solutions to register a picture against a reference in the translation case [MAI 87 a], the parametric case [WU 86] and in the non-parametric case [MAI 87]. In all these cases dynamic programming finds a global matching optimum as the best path between two sequences of features. Applying also the technique of dynamic programming Gormann and al. [GOR 86] use contour segments characterised by their Fourier descriptors to recognize partially distorded or incomplete contours.

This paper presents a new method for recognizing distorded or occluded objects in an image. The objects are assumed to be flat. The method uses line segments to describe a given shape. Each segment is characterised by its length l, its midpoint coordinates (x, y) and its orientation θ. We proceed through a table were the columns represent the M model segments and the rows the N scene segments. The basic idea is to generate in this table diagonal paths, each one representing a matching between a part of the model and a part of the scene. The longest path is considered as the best match.

2 The matching procedure

2.1 Generating paths

The paths generated in the $M \times N$ table consist of nodes corresponding to local matches between scene and model segments. Each path corresponds to a match between model and scene contour parts. At each node a model to scene transformation and a cost value are estimated. The transformation is the product of a rotation, a scaling and a translation. As we develop a path we calculate the average of all the local transformations corresponding to the path. The cost value permits us to control the quality of the match, and guarantees that, for a given path, the variation between the averaged transformation and the local transformations is not too important.

Let's suppose that we have developped, up to the node $(i-1, j-1)$, a path representing a partial match. At this node we store the parameters of the corresponding averaged geometrical transformation $\overline{T}_{i-1,j-1}$, the cost $c(i-1, j-1)$ associated with the path, and the length of the path $L(i-1, j-1)$ which is the number of nodes of the path. At the node (i,j) we compute the parameters corresponding to a local matching between the scene segment S_i, $S_i = (x_i^s, y_i^s, l_i^s, \theta_i^s)$, and the model segment M_j, $M_j = (x_j^m, y_j^m, l_j^m, \theta_j^m)$. This matching is expressed by a geometrical transformation $\widehat{T}_{i,j} = (\widehat{k}_{ij}, \widehat{\theta}_{ij}, \widehat{t}_{ij}^x, \widehat{t}_{ij}^y)$, $\widehat{k}_{ij}, \widehat{\theta}_{ij}, \widehat{t}_{ij}^x$ and $\widehat{t}_{ij}^y$ being respectively the scale, the rotation and the two translation parameters. These parameters are computed by:

$$\widehat{k}_{i,j} = l_j^s / l_i^m \quad (1)$$

$$\widehat{\theta}_{i,j} = \theta_i^s - \theta_j^m \quad (2)$$

$$\widehat{t}_{i,j}^x = x_i^s - \widehat{k}_{ij}.(x_j^m.\cos\widehat{\theta}_{ij} - y_j^m.\sin\widehat{\theta}_{ij}). \quad (3)$$

$$\widehat{t}_{i,j}^y = y_i^s - \widehat{k}_{i,j}.(x_j^m.\sin\widehat{\theta}_{ij} + y_j^m.\cos\widehat{\theta}_{ij}). \quad (4)$$

Then we compare $\widehat{T}_{ij}$ to the averaged transformation $\overline{T}_{i-1,j-1}$, by computing a distance value $d(\overline{T}_{i-1j-1}, \widehat{T}_{ij})$. This distance measure is a weighted sum of the absolute differences between the parameters of the two transformations:

$$\begin{aligned} d(\overline{T}_{i-1j-1}, \widehat{T}_{ij}) = \; & \tfrac{a}{\Delta k_m}.\mid \widehat{k}_{ij} - \overline{k}_{i-1j-1} \mid \\ & + \tfrac{b}{\Delta\theta_m}.\mid \widehat{\theta}_{ij} - \overline{\theta}_{i-1j-1} \mid \\ & + \tfrac{c}{\Delta t_m^x}.\mid \widehat{t}_{ij}^x - \overline{t}_{i-1j-1}^x \mid \\ & + \tfrac{c}{\Delta t_m^y}.\mid \widehat{t}_{ij}^y - \overline{t}_{i-1j-1}^y \mid \end{aligned} \quad (5)$$

$d(\overline{T}_{i-1j-1}, \widehat{T}_{ij})$ takes a minimum value of zero when the two transformations are identical, and increases with the disparity of the transformations parameters. With the distance d and the cost value $c(i-1, j-1)$ associated with the path ending at the node $(i-1, j-1)$, we evaluate the cost $c(i,j)$ of the path we plan to extend to the node (i,j). $L(i-1, j-1)$ being the path length, we have :

$$c(i,j) = \frac{c(i-1,j-1).L(i-1,j-1) + d(\overline{T}_{i-1j-1}, \widehat{T}_{ij})}{L(i-1,j-1)+1} \quad (6)$$

If $c(i,j)$ is within a given threshold ϵ the path is extended to the node (i,j). We must therefore evaluate the parameters of the mean geometrical transformation relative to the extended path. Considering that the path begins at the column n, these parameters are computed as follows :

$$\overline{k}_{ij} = \frac{\overline{k}_{i-1j-1}.\sum_{p=n}^{j-1} l_p^m + \widehat{k}_{ij}.l_j^m}{\sum_{p=n}^{j} l_p^m} \quad (7)$$

$$\overline{\theta}_{ij} = \frac{\overline{\theta}_{i-1j-1}.\sum_{p=n}^{j-1} l_p^m + \widehat{\theta}_{ij}.l_j^m}{\sum_{p=n}^{j} l_p^m} \quad (8)$$

$$\overline{t}_{ij}^x = \frac{\overline{t}_{i-1j-1}^x.\sum_{p=n}^{j-1} l_p^m + \widehat{t}_{ij}^x.l_j^m}{\sum_{p=n}^{j} l_p^m} \quad (9)$$

$$\overline{t}_{ij}^y = \frac{\overline{t}_{i-1j-1}^x.\sum_{p=n}^{j-1} l_p^m + \widehat{t}_{ij}^y.l_j^m}{\sum_{p=n}^{j} l_p^m} \quad (10)$$

We store the cost $c(i,j)$ and all these parameters in the table entry (i,j). In the case where $c(i,j)$ is greater than the threshold ϵ the path ends at the node $(i-1, j-1)$ and a new one begins at the node (i,j).

Furthermore, to treat the case of closed contours, we try to extend each path reaching a node (i, N), i.e the last column, to a node $(i+1, 1)$. This is available only for the paths which do not begin at the first column.

2.2 Finding new matches

Once all the paths have been generated we consider the longest one to which an averaged transformation $\overline{T}$ and a cost c are associated with. The transformation $\overline{T}$ is therefore considered as the model to scene geometrical transformation. Because of the partial occlusions the path may have been interrupted. In order to find other matches, we apply the transformation $\overline{T}$ to the model segments outside the path, and for each one, we try to find the corresponding scene segment. Hence any of the model segment $M_j = (x_j^m, y_j^m, l_j^m, \theta_j^m)$ is transformed into a segment $M_j^* = (x_j^*, y_j^*, l_j^*, \theta_j^*)$ by the transformation $\overline{T} = (\overline{k}, \overline{\theta}, \overline{t}^x, \overline{t}^y)$. The image of M_j is given by :

$$x^* = \overline{t}^x + \overline{k}.(x_j^m.\cos\overline{\theta} - y_j^m.\sin\overline{\theta}) \quad (11)$$

$$y^* = \overline{t}^y + \overline{k}.(x_j^m.\sin\overline{\theta} + y_j^m.\cos\overline{\theta}) \quad (12)$$

$$\theta^* = \theta_j^m + \overline{\theta} \quad (13)$$

$$l^* = \overline{k}.l_j^m \quad (14)$$

For each unmatched segment S_i we compute a dissimilarity measure d_{ij} between M_j and S_i. This measure is a positive weighted sum of the differences between the segments' parameters and is similar to the one used in [AYA 86]:

$$d_{ij} = p.\frac{\Delta\theta_{ij}}{\Delta\theta_{max}} + q.\frac{\Delta E_{ij}}{\Delta E_{max}} + r.\frac{\Delta L_{ij}}{\Delta L_{max}} \quad (15)$$

p, q and r are the positive weights and $\Delta\theta_{max}$, ΔE_{max} and ΔL_{max} are the upper tolerances for each difference value. The ratio providing a more reasonable comparison for the length, we define ΔL_{ij} as :

$$\Delta L_{ij} = \frac{\mid l_j^* - l_i^s \mid}{l_i^s} \quad (16)$$

On the other hand, an absolute value of the differences between the angles or the midpoint coordinates is sufficiently relevant. Then $\Delta\theta_{ij}$, ΔE_{ij} are defined as :

$$\Delta\theta_{ij} = \mid \theta_j^* - \theta_i^s \mid \quad (17)$$

$$\Delta E_{ij} = ((x_j^* - x_i^s)^2 + (y_j^* - y_i^s)^2)^{\frac{1}{2}} \quad (18)$$

The most similar candidate S_i is the one giving the minimum value d_{ij} which is also lower than one. At this stage of the algorithm we evaluate anew the parameters of the segment M_j to the segment S_i transformation by the equations (1) (2) (3) (4), and we compute $d(\overline{T}, \widehat{T}_{ij})$ with (5). We then evaluate the cost $c(i,j)$ by (6). In the case where it is lower than ϵ, M_j is matched with S_i and the parameters of the mean transformation are adjusted with equations (7) (8) (9) and (10). Otherwise M_j is left without scene correspondent and we consider a new model segment. The search process stops with the last model segment explored. The last step consists of the evaluation of the identification quality Q. Q is defined as a percentage of the model length recognized.

3 Complexity analysis

Let us consider the problem of finding the longest substring of scene segments corresponding to a substring of model segments for a well precise transformation. A simple algorithm to treat this problem is to shift the segment string of the model all along a segment string of the scene looking for a match and to repeat the process for all the scene segment strings. If a local estimation of the transformation is used as in the algorithm presented in this paper, . The complexity of this algorithm is $O(N \times M^2)$. M being the number of model segments and N the number of scene segments. The algorithm presented in this paper is constituted of two main steps. The first one consists of finding diagonal paths in a $M \times N$ table by proceeding line by line sequentially. And the second one consists of finding the remaining local correspondences outside the longest path. Thus the time complexity of the algorithm is $O(N \times M)$. It has the main advantages to be easily parallelized on a specialized hardware, to be fast and to require a minimum memory space. In fact we can proceed diagonally through the table. Hence we need to keep in memory the informations relative to the two last nodes of the path being generated, and the informations relative to the last best path. sequential, we need to keep in memory only two rows of the table to estimate a transformation and to try to extend a path. That is because at a node (i, j) we need the informations stored at the node $(i-1, j-1)$.

4 Experimental results

The algorithm has been implemented in C language on a DN-3000 APOLLO work station. All the segments were obtained experimentally. The images were aquired by a C.C.D camera in normal lighting conditions. We obtained images of size 256*256. The noise was then suppressed and the contours found by applying a Sobel operator and a thresholding technique. These contours were thinned and we applied the Giraudon technique [GIR 87] to find a list of pixel chains representing the contours. Each pixel chain was approximated to line segments by using the Pavlidis technique [PAV 82]. To improve the performances of the algorithm we had to choose carefully all the weight factors used with the distance and the dissimilarity computations (the distance between two transformations and the dissimilarity between two segments). Because no rules have been established for that purpose, we chose the best weigths parameters experimentally.

*Fig*1 shows a composite scene of a hammer and pliers with no overlap.This scene consists of 57 segments. The hammer shown in *Fig*3 is used as a model and is composed of 44 segments. The hammer is transformed by the transformation found at the end of the path during the matching process. *Fig*4 shows the result of the superposition of the transformed model upon the scene.

*Fig*2 is an image of the hammer overlapped by pliers. This scene has 67 segments. *Fig*5 gives the result of the superposition of the model transformed upon the scene. The time of the matching is globally under 20 seconds.

We can see through these two examples, that the transformations found after the matching process give a quite accurate position of the model into the scene.

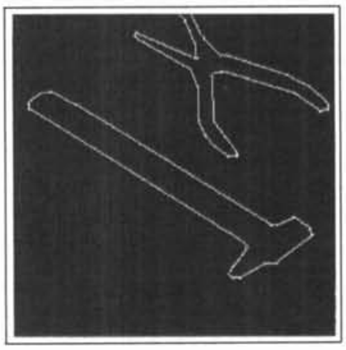

Figure 1: Scene A

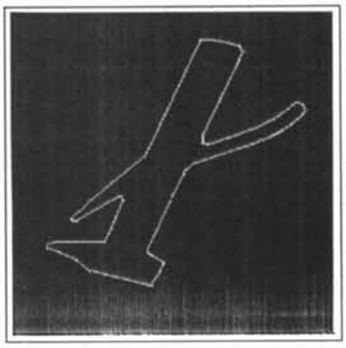

Figure 2: Scene B

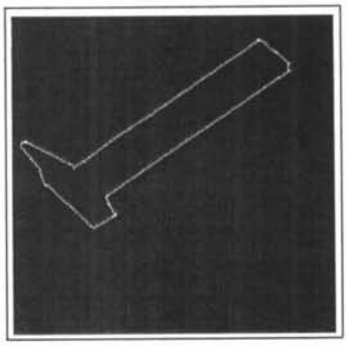

Figure 3: The model

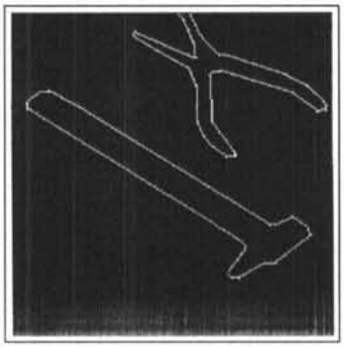

Figure 4: The transformed model superposed upon the scene

Figure 5: The transformed model superposed upon the scene

5 Conclusion

In this paper we have proposed a new method to try to solve the matching problem between a scene and a model. The algorithm uses line segments which describe the boundary shapes and proceeds as follows : through a table were the columns represent the model segments and the rows the scene segments, diagonal paths are generated. These paths consist of nodes corresponding to a local matching between a scene and a model segment. Each path represents a matching between a part of the model and a part of the scene. We consider that the longest path corresponds to the best match. We then try to find other consistent matches outside this path. The algorithm has been shown to recognize unknown contours which have been rotated and translated. Furthermore precise knowledge of scale is not required.However the algorithm also suffers from some limitations. A problem is encountered when a model segment appears fragmented in the scene. In this particular case the model segment corresponds to two or more shorter segments in the scene. One can try to solve the problem by considering vertical paths in the table. These vertical paths would express the fact that some successive scene segments correspond to one model segment. However for this solution we must find robust similarity and cost measures. The other solution we can look at is the line linking : short lines could be linked to form a longer one. With this solution, in addition to the increase of the computational time, we must be careful not to over-link. The study of these two solutions to the problem of broken lines will be the subject of our future works.

References

[AYA 86] : N. Ayache, O.D. Faugeras,"HYPER: A new approach for the recognition and positioning of two dimensional objects." IEEE Trans. Pattern. Anal. Mach. Intell. vol. PAMI-8 pp.44-54 Jan. 1986.

[BAL 82] : D.H. Ballard, C.M. Brown, "Computer vision.",Prentice-Hall 1982.

[BHA 84] : B. Bhanu, O.D. Faugeras, "Shape matching of two-dimensional objects." IEEE Trans. Pattern. Anal. Mach. Intell. vol. PAMI-6 No 2 pp.137-156 Mar. 1984

[BOL 82] : R.C. Bolles, R.A. Cain, "Recognizing and locating partially visible objects: the local feature focus method." The international journal of robotics research vol. 1, No 3, pp 57-82, 1982.

[BOU] : E.H. Bouyakhf, "A new approach for recognition and locating of industrial parts." L.A.A.S. Rapport de recherche L.A.A.S.

[GIR 87] : G. Giraudon, "Chainage efficace de contour." I.N.R.I.A, Rapport de Recherche No 605 France, Fev. 1987.

[GOR 86] : J.W. Gorman, O.R. Mitchell, F.P. Kuhl, "Partial shape recognition using dynamic programming" IEEE Trans. Pattern. Anal. Mach. Intell. vol. PAMI-10 pp.257-266 Mar. 1988

[LEE 90] : H. LEE, D. Yu, "Line based structural matching via segment splitting.", Pattern Recognition Letters vol. 11, No 3, pp.181-189, Mar. 1990.

[MAI 87 a] : H. Maitre, Y. Wu, "Improving dynamic programming to solve image registration.", Pattern Recognition, vol. 20, No 4, pp 443-462, 1987.

[MAI 87 b] : H. Maitre, Y. Wu, "Combination of Dynamic Programming and autoregressive model for elastique matching of distorted picture.", Proc. 5th Scandinavian Conference on Image Analysis, pp.109-116,Stockholm,June 1987.

[MAI 88] : H. Maitre, Y. Wu, "Elastic matching versus rigid matching by use of dynamic programming.", Proc. IEEE 9th International Conference on Pattern Recognition, pp. 79-81, Rome, Nov. 1988.

[McIN 88] : J.H. McIntosh, K.M. Mutch, "Matching straight lines.", Computer Vision, Graphics, and Image Processing 43, pp 386-408, 1988.

[PAV 82] : T. Pavlidis, "Algorithms for graphics and image processing. ", Springer Verlag, Berlin 1982

[SKO 88] : T. Skordas, R. Horaud, "Mise en correspondence structurelle pour la vision stereoscopique.", Technique et Science Informatique vol. 7, No 6, pp. 591-607, 1988.

[WU 86] : Y.F. WU, H. Maitre, "Registration of rotated picture with hidden parts using dynamic programming.", Proc. IEEE 8th International Conference on Pattern Recognition, pp. 792-794, Paris, Oct. 1986.