

# Recognizing Partially Visible 2-D Non-rigid Wire-Shapes\*

J. H. Sossa-Azuela and A. E. de León-Gutierrez

Centro de Investigación en Computación-IPN

Unidad Profesional Adolfo López Mateos

México, D.F. 07738, México

E-mail:hsossa@pollux.cic.ipn.mx

**Abstract.** A technique for the fast recognition of partially visible 2-D non-rigid wire-shapes is presented. For this, the skeleton of an object is first decomposed into a set of parts called fundamental parts (FPs). Each FP is next described by a code composed of five integer numbers. The code of each FP is then used to build the corresponding database of models. During a second phase (of indexing), the FPs extracted from a test image are used to index the model database subtracting from it the list of objects possibly present in the image. The performance of the system is tested with a set of insect designs.

## 1. Introduction

The recognition of partially visible 2-D shapes is of particular interest in areas such as microscopy. In a previous work [1] we presented a technique useful for the modeling and recognition of isolated planar rigid and non-rigid shapes. In this work, that technique is extended to the case of non-rigid shapes tacking into account now the presence of occlusions and lack of information.

The main difference between the proposed technique in this work and the one reported in [1] is based on the fact that to recognize an object, it is necessary to decompose the object into a set of parts called *fundamental parts* (FPs). These FPs, are obtained from the skeleton of the object, considering each crossing point as a generating point, and by adding to this point the nearest terminal points of crossing points as we will see in the next sections.

The set of features used to model the objects, to build the database of models and to recognize the objects is the same used in [1], i.e. the number of terminal points (points with a neighbor, TPs), and the number of three-edge points (points with three neighbors, TEPs). Additionally, to these features we have added three of the coefficients of the second immanantal polynomial of the Laplacian

---

\*This research was supported by the Centro de Investigación en Computación of the IPN (CIC-IPN), the Consejo Nacional de Ciencia y Tecnología of Mexico (CONACYT), and the Centro de Investigación y de Estudios Avanzados of the IPN (CINVESTAV-IPN).

matrix of the graph associated to the FP (see [2]). The reason to extend the number of features is to facilitate differentiation among objects, enhancing thus the performance of the whole system.

The decomposition of an object into a set of FPs will allow us, as we will see in the section of experiments, the recognition of possibly occluding non-rigid planar wire-shapes taking advantage, in one hand, the redundancy of the FPs, and, in the other hand, their locality.

## 2. The Technique

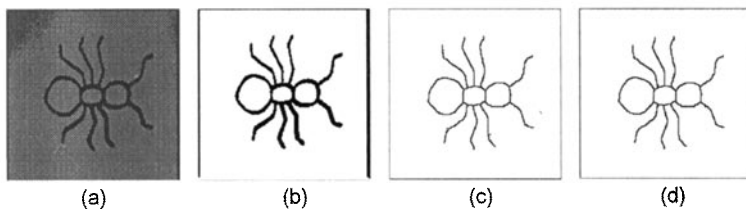
The technique is divided into two phases: model database construction and indexing. Both phases share seven steps: preprocessing, skeletonizing, skeleton pruning, connectivity graph generation, parasite point fusion, FP generation, and FP codification. Each of these steps is next explained.

### 2.1. Model Database Construction

Without lost of generality, and for the sake of simplicity, we will suppose that the image contains just one object.

#### 2.1.1. Preprocessing

During this step a gray level image is manually or automatically thresholded. In this case a standard thresholding method is used. Figure 2.1 (b) shows the binary version of the image shown in figure 2.1 (a).



**Fig. 2.1:** (a) Gray level image of an insect's desing (b) Binary image. (c) Sskeleton. (d) Pruned skeleton.

#### 2.1.2. Esqueletonizing

The image obtained by thresholding is now skeletonized to obtain the skeleton or medial axis of the object in the image. A 4-skeletonizer [3] was used here because as was shown in [3] a 4-connected skeleton allows to compute, with any problem, the desired features (the TPs and the TEPs). Figure 2.1 (c) shows the resulting 4-connected skeleton for the object of figure 2.1 (b) when this method is applied.

### 2.1.3. Skeleton Pruning

The skeleton produced by any skeletonizing algorithm sometimes comes with some small parasite branches (for an example, see figure 2.1 (c)). Saliences like these cause that the number of TPs and TEPs and the coefficients of the second immanantal polynomial change, affecting thus the description of the associated FP. A pruning process of these branches is then needed. All branches smaller than a specified threshold  $P$ , are thus reduced. Figure 2.1 (d) shows the pruned skeleton corresponding to the skeleton of figure 2.1 (c). The value for  $P$  was  $P = 5$  pixels.

### 2.1.4. Connectivity Graph Generation

From the pruned skeleton it is possible to obtain the corresponding connectivity graph (CG). This graph tells us which terminal points or crossing points are connected to which terminal point and/or crossing points in the skeleton. To obtain this graph, the image is scanned until the first terminal point or crossing point is reached. The skeleton is next followed through its different branches (depending on the point's valency) using a contour follower, labelling the branches and the terminal or crossing point reached during the process. This process is repeated for the remaining TPs and TEPs. At the end of this process each TP or each TEP and their corresponding edges will be completely labeled.

### 2.1.5. Crossing Point Fusion

The right decomposition of an object into its FPs is primordial to the well performance of the system. Such FPs, as was said in the last section, are obtained from the CG of the object' skeleton. A FP is formed from each crossing point in the connectivity graph by adding to that point the nearest terminal or crossing points (see section 2.1.6 for more details). Most of the times, during image processing, a point appears decomposed into crossing points of less order [4]. This will affect the FP generation step, deteriorating in consequence the system performance. To counterattack this problem, inherent to any digitalization process we used a fusion mechanism in order to robust the FP generation module. This module takes as input the CG of the of an object' skeleton. For each pair of crossing points, the module verifies if the size of the branch joining these points is less than a threshold, in that case, the points are considered as one, i.e. they are fused.

### 2.1.6. Fundamental Part Generation

At the beginning of section 1 it was said that a non-rigid object, in the content of this work, is recognized in terms of its FPs. These are obtained form the CG associated to the skeleton of the object, considering each crossing point as an "origin" and adding to it the nearest terminal and crossing points. Figure 2.2 shows the FPs of size two for the pruned skeleton of figure 2.1 (d).

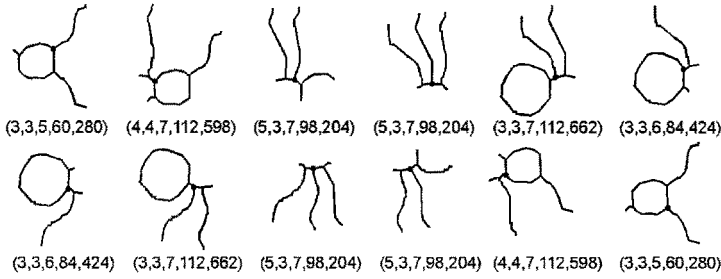


Fig. 2.2: Fundamental parts of size 2 and their codes for figure 2.1(d).

**2.1.7. Fundamental Part Codification**

To each FP is given a code composed by five integers. The number of TPs, the number of TEPs and the coefficients number one, two and before last of the second immanantal polynomial of the Lplacian matrix of the graph associated to the FP. The TPs, and TEPs, and the coefficients of the second immanantal polynomial were already used successfully, respectively, in the recognition of isolated rigid and non-rigid shapes [1] and in the indexing of polyhedral objects [2].

Just to remember the number of TPs and the number of TEPs are, respectively, the number of terminal points and the number of three-edge points in the FP. Remember also that the number of TEPs are obtained from the crossing points in the FP, knowing that a crossing point can be decomposed into its TEPs, and that this number, as was said in [4], equals the number of branches composing the crossing point -2.

In the case of the coefficients first, second and before last of second immanantal polynomial we proceed as follows. In the content of this work each FP can be seen as a graph, say  $G$ . The coefficients  $c_0, \dots, c_n$  of the second immanantal polynomial of the  $n \times n$  Laplacian matrix,  $L(G)$  [6] associated to  $G$ , are integers and they can be computed as follows [5]. If  $n$  and  $m$  are, respectively, the number of nodes and the number edges of graph  $G$ , then:

$$\left\{ \begin{array}{l} c_0(L(G)) = n - 1 \\ c_1(L(G)) = 2m(n - 1) \\ \vdots \\ c_k(L(G)) = \sum_{X \in Q_{k,n}} \left( \sum_{i=1}^n l_{ii} \det(L(G)\{X\}(i)) - \det(L(G)\{X\}) \right) \end{array} \right. \tag{2.1}$$

In these formulae  $l_{ii}$  denotes a diagonal term of  $L(G)\{X\}(i)$  and  $Q_{k,n}$  denotes the set of all the  $C_n^k$  strictly increasing sequences of size  $k$  ( $2 \leq k \leq n$ ) obtained from the set  $\{1, 2, \dots, n\}$ . For any  $n \times n$  matrix  $M$  and for  $X \in Q_{k,n}$   $M[X]$  is the  $k \times k$  principal submatrix of  $M$  corresponding to  $X$ .  $M\{X\}$  is the  $n \times n$  matrix:

$$M \{X\} = \begin{pmatrix} M [X] & 0_k \\ 0_k & I_{n-k} \end{pmatrix} \quad (2.2)$$

where  $I_{n-k}$  is the identity matrix of size  $n - k$  and  $0_k$  is the null matrix of size  $k$ .  $M \{X\} (i)$  is the matrix obtained from  $M \{X\}$  by removing the  $u$ th row and the  $i$ th column.

As we said, in the content of this work, only the first, second and before last coefficients, i.e.  $c_0, c_1$  and  $c_{n-1}$  will be used to describe each FP.

It is known that the Laplacian matrix is only applied to simple graphs, i.e. graphs without loops and where two vertices are joined just by one edge. If the graph associated to a FP presents a loop, when the FP is generated, the loop is first decomposed into three edges by adding to it two internal vertices before the FP is coded. This is the case of FPs 5 and 8 of figure 2.2. Similarly, if the graph presents two vertices joined by two edges, one of them is decomposed into two edges by adding to it on internal point, again before the FP is coded. This is the case of FPs 6 and 7 of figure 2.2. This way, the FP's complex graph is reduced to a simple one. Figure 2.2 shows the set of FPs of the object of figure 2.1 (d) and their codes.

### 2.1.8. Model Database Construction

As mentioned at the beginning of section 2, the 2-D non-rigid shape identification process is divided in two phases: model database construction and indexing. In sections 2.1.1 to 2.1.7, the different algorithms describing the functioning of the common steps of both phases were presented. During model database construction, the image is first binarized and skeltonized. The resulting skeleton is then pruned, and the CG obtained. Parasite crossings are next fusioned. FPs are next generated and coded. These codes are finally used to build the database of models.

At the end of this process the corresponding model database will be organized. In practice, it can take several forms: an ordered list of codes (the codes of each FP) with pointers to the set of objects where the corresponding FP are located, or a hash table, whose buckets are, in this case, the addresses obtained by transforming each PF's code through a suitable hashig function. As in the case of a ordered list, each address in the hash table will contain a pointer to the list of objects containing the corresponding FP. In this work, the first data structure was chosen. The second one is normally applied when the number of object to be indexed is big and when we want to obtain very fast accesses to the database of models.

## 2.2. Indexing

During this phase, the set of FPs extracted form a test image, containing one or more possibly occluding shapes is used to index the previously constructed model database, retrieving from it the list of object possibly present in the image. For this, the next two steps are applied.

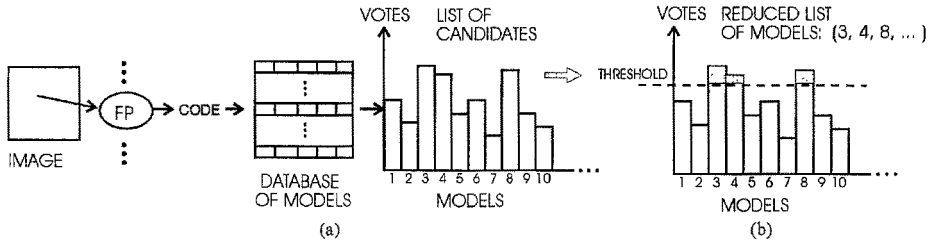


Fig. 2.3: (a) Candidate selection. (b) Candidate reduction.

### 2.2.1. Candidate Selection

During this step, the code of each FP in the test image is compared with the codes in the model database. If they match, the corresponding list of objects containing this FP is retrieved. At the end of this process (see figure 2.3(a) for an example) we will have a set of hypotheses (a list of object for each FP found in the test image) arranged as a histogram of votes. The number of votes is the number of times an object was indexed.

### 2.2.2. Candidate Reduction

It is clear that at the end of the last process, we will have a histogram containing the number of votes an object received during candidate selection (see figure 2.3(b)). Some models will have more votes than others. It is thus necessary to decide which models are most suitable to be present in the image. Intuitively, those with the bigger number of votes are the best candidates. For this a reduction threshold  $T$  was chosen. Those candidate models whose number of votes is above the threshold will be the final models (see figure for an example). The value for  $T$  in the experiments was chosen as the arithmetic mean of the votes received by each object during the candidate selection process. All votes were equally weighted, its value was 1.0.

## 3. Experimental Results

In this section, some experimental results are shown. To test the performance of the proposed technique, the set of objects shown in figure 3.1 was used. This kind of objects was chosen due to its beauty and variety.

### 3.1. Model Database Construction

To each of the twelve objects shown in figure 3.1, the set of methods described in sections 2.1.1 to 2.1.7 were applied to obtain the codes of the FPs of the corresponding skeleton. These codes were next used to build the database of models. As was said in section 2.1.8, it is a list of keys stocked in an ordered way. To each key (the FP's code) a list of the associated objects is given.

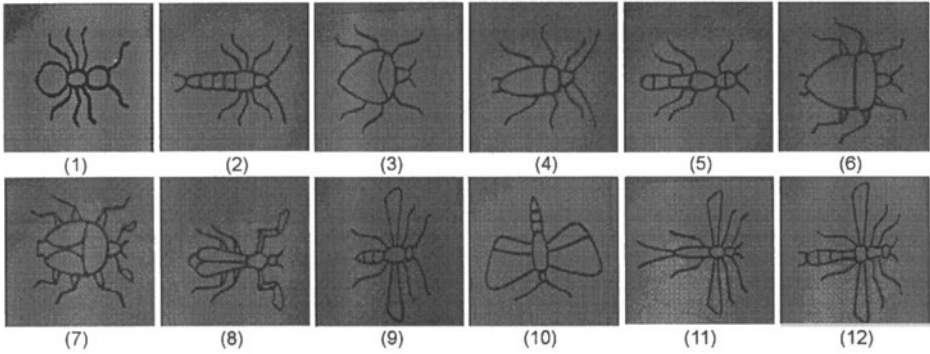


Fig. 3.1: The set of testing objects.

### 3.2. Indexing

The performance of the proposed technique was tested with several sequences of images containing one or more objects in the presence of small occlusions. Each sequence was obtained by choosing and manually designing over paper some of the insects shown in figure 3.1 by taking the corresponding images by means of a B/W CCD camera. One of the sequences is shown in figure 3.2. Note how in some cases, some of the insects appear complete; in other cases they appear incomplete. To each one of these images, the sequence of methods described in 2.1.1 to 2.1.7 was applied. Note, also that the number of objects to be analyzed in each image is, respectively, 3, 2, 2, and 2, for a total of 9.

The results are shown in Table 3.1. The chosen selection threshold was automatically chosen as specified in section 2.2.2. Note that in the nine cases, the object looked for was found (underlined). Notice however, that together with the object, other models were also indexed. In all cases, the models are arranged in descending order according to number of votes received.

If we consider as "acceptable" the fact that in each of the lists of models outputted by the system appear the objects searched for, we can consider that the system's performance is satisfactory.

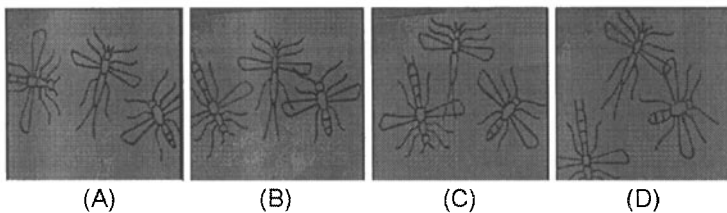


Fig. 3.2: One of the testing sequences.

Image No.	object 1	object 2	object 3
A	<u>11</u> ,12,4,1,3,5	<u>12</u> ,9,5,11,1,2,3,7	2,12,5, <u>9</u> ,1,3,10
B	<u>9</u> , <u>11</u> ,8,12,7,5,6	<u>12</u> ,2,5,11,1,3,10	-
C	<u>12</u> ,2, <u>11</u> ,7,4,5	<u>9</u> ,12,1,2,5,8,10	-
D	<u>9</u> , <u>11</u> ,12,5,4,8	<u>12</u> ,2,1,3,5,11	-

Table 3.1: Indexing results during candidate selection.

## 4. Conclusions

A new technique for the recognition of partially visible 2-D non-rigid wire-shapes was presented here. During the first phase, each object is first decomposed into a set of parts called fundamental parts (FPs). These are obtained from the skeleton of the object. To each of these FPs is associated a code composed of five integers: the number of terminal points, the number of three-edge points, and three coefficients of the second immanantal polynomial of the Laplacian matrix of the graph associated to the FP. The code of each FP is next used to build the corresponding model database. It is a ordered list of 5 dimensional vectors with pointers to the objects where the FP is found.

During the second phase, the set of FPs extracted from a test image containing one or more possibly occluding shapes, are used to index the database of models previously constructed, subtracting from it the list of objects most possible present in the image. For this a voting and thresholding mechanism was used. The performance of the system was tested with a set of insect designs.

## References

- [1] J. H. Sossa and J. Díaz-de-León S., Recognizing 2-D rigid and non-rigid wire-shapes, in *Advances in Structural and Syntactical Pattern Recognition, Lecture Notes in Computer Science*, 1121. Springer, 300-309, 1996.
- [2] R. Horaud and J. H. Sossa, Polyhedral object recognition by indexing, *Pattern Recognition*, 28(12), 1995, pp. 1855-1670.
- [3] J. Díaz-de-León S., Skeletonizing algorithms for binary images, Master's Thesis, CINVESTAV-IPN. Electrical Engineering Department, Mexico, D.F., August de 1993. In Spanish.
- [4] J. L. Díaz de León S. and J. H. Sossa, On the computation of the Euler number of a binary object. *Pattern Recognition*, 29(3), 1996, pp. 471-476.
- [5] R. Merris, The second immanantal polynomial and the centroid of a graph, *SIAM J. Alg. Disc. Meth.* 7, 1986, pp. 484-503.
- [6] G. M. Constantine, Graph complexity and the laplacian matrix in blocked environments, *Linear and Multilinear Alg.* 28, 1990, pp. 49-56.