

Recognizing 2-D Rigid and Non-rigid Wire-Shapes*

J. Humberto Sossa A.^{1†} and J. Luis Díaz-de-León S.²

¹Centro Nacional de Cálculo-IPN

Unidad Profesional Adolfo López Mateos S/N, México D.F. 07738, México.

²CINVESTAV-IPN, Depto. de Ingeniería Eléctrica, Secc. de Control Automático
Av. IPN # 2508, México, D.F. 07300, México.

Abstract. A new method for the fast recognition of two-dimensional wire-shapes is presented. Both the rigid and the non-rigid case are explored. The proposed approach is based on a characterization of the skeleton of the shape in terms of a set of two topological robust features: *terminal points* (TPs), i.e. points with just one neighbor and *three-edge-points* (TEPs), i.e. points with only three neighbors. The number of TPs and TEPs in the skeleton are used as inputs to look up the object's database for the fast recognition of the shape. The proposed technique profits the advantages of the *Fast Distance Transformation* (FDT) to obtain rapidly the skeleton. These two characteristics make of the proposed approach a fast and simple method for the fast recognition of 2D binary objects, which is desired for real time applications.

1. Introduction

In the field of image processing and computer vision, object recognition plays an important role. Indeed, for a robot manipulator as well as for a mobile robot, identification and location of objects in their working environment are principal steps allowing them to perform tasks such as picking, assembly and obstacle avoidance.

The identification problem consists on assigning names or labels to each of these objects. Two main approaches have been used to solve this problem. In the first approach, a model of each object in the model database is matched against the scene image and a score is calculated [11]. The unknown object is recognized by the highest score. In the second approach, in order to eliminate the process of matching every model with the input scene, local features are used to generate a hypothesis to see if the model can be found in the scene image [5]. This hypothesis is then usually verified by a point to point matching of a template of the object with the image.

*This research was supported by the National Council of Science and Technology of México and the Mexican Institute of Communications.

[†]Humberto Sossa is also a researcher at CINVESTAV-IPN, Depto. de Ing. Eléctrica, Av. IPN. 2508, México, D.F. 07300, México, E-mail: hsossa@ctrl.cinvestav.mx

In this work, the object identification problem is solved by means of a topological characterization using the skeleton of the shape for it preserves the structural information of the shape. We are then restricted to the case of isolated objects that can change their geometry but not their topology (see Figure 1.1). A 4-skeleton is chosen in this case because as it was shown in [1], 4-skeletons do not present the inconvenient of parasite holes, effect that is sometimes present in 8-skeletons after skeleton obtaining (Figure 1.2). The topological features used here are the *terminal points* (TPs), i.e. points with just one neighbor and the *three-edge-points* (TEPs), i.e. points with only three neighbors. TPs and TEPs are used here because they have proven to be very nice features to obtain the Euler Number or *genus* of a shape [2]. The number of TPs and TEPs in the object's skeleton are used as inputs to look up the object's database to rapidly identify the shape. To substantially speed the identification process, the skeleton of the object is obtained by means of the recent algorithm proposed in [1], which profits the advantages of the *Fast Distance Transformation* (FDT) to rapidly obtain the desired skeleton. This methodology will be applied for the recognition of both rigid and articulated wire-shapes.

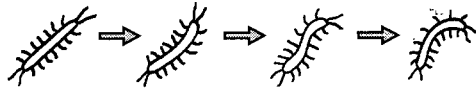


Fig. 1.1: An object with different geometry but with the same topology.

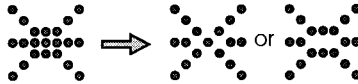


Fig. 1.2: Pathological cases sometimes present in 8-skeletons.

In section 2 the two topological features used in this work: the terminal point (TP) and the three-edge-point (TEP), are first presented. They were already formally defined in [13] in terms of the so called *Yokoi's connectivity numbers* and after in [10] in terms of the *Rutovitz's crossing number* [9]. In [2], it was also shown that any crossing point, i.e. a point with three or more neighbors, can be decomposed into a set of TEPs and this number equals $n - 2$, with n the number of branches. This result is important because as we will see in this paper it will allow us to define a formal methodology to describe any 2-D wire-shape using its skeleton. Section 3 describes how TPs and TEPs are obtained and how they are used to describe 2-D shapes. In section 4 the identification approach is presented. This last takes as input the shape's description extracted from each one of the objects in an image to obtain the object's identity. In section 5 a set

of shapes (both rigid and flexible) is then used to test the performance of the system. Finally, in section 6 some conclusions and some further work are given.

2. TPs, TEPs and Crossings

In this section the two topological features used hereafter in this work: TPs and TEPs, are presented. As we said, they were already formally defined in [13] and [10]. We will also recall how any crossing point, i.e. a point with three or more neighbors, can be decomposed into a set of TEPs and this number equals $n - 2$, with n the number of branches.

Definition 1. Let $q = (x, y)$ and $p = (u, v)$ with $q, p \in Z^2$ and $K \in \{4, 8\}$, then the following metrics are widely used:

$$\begin{aligned} d_4(q, p) &= |x - u| + |y - v| \\ d_8(q, p) &= \max(|x - u|, |y - v|) \end{aligned}$$

Hereafter, these metrics will be denoted by d_4, d_8 (or d_K where K is the basis of the metric).

Definition 2. Let (A, α) a bidimensional metric space and α any metric, and let $N(p) = \{p_i \mid 0 < \alpha(p_i, p) \leq 1\}$ for A discrete or $N(p) = \{p_i \mid \alpha(p_i, p) \rightarrow 0 \text{ but not } 0\}$ for A real, the neighborhood of p and N_p its cardinality, i.e. the number of neighbors of p different from 0. Then:

1. If $N_p = 1$ we say that p is a terminal point (TP).
2. If $N_p = 2$ we say that p is an internal point.
3. If $N_p \geq 3$ we say that p is a crossing point or simply a crossing.
4. If $N_p = 3$ we say that p three-edge-point (TEP).

Definition 3. For (Z^2, d_k) with $d_k \in \{d_4, d_8\}$, we accept as a skeleton any K -connected set of points Q for which if a non-TP point is removed, the set Q is decomposed into two or more K -connected sets.

Notice that for our definition of skeleton, the number of branches crossing a node in a 3×3 neighborhood is almost always the cardinality N_p of p (Figure 2.1 (a,b)), except for some particular cases (a few particular cases) such as the ones shown in Figure 2.1 (c).

For these particular cases, instead of having one point as the point joining several branches we have four. Although these configurations may appear in real cases, they can be easily solved by special masking [1].

Another characteristic of 3×3 discrete neighborhoods is that non-particular cases containing more than 4 branches are not possible. The admissible maximal non-particular cases after a good skeletization (unitary skeletons) are the ones shown in Figure 2.1 (d).

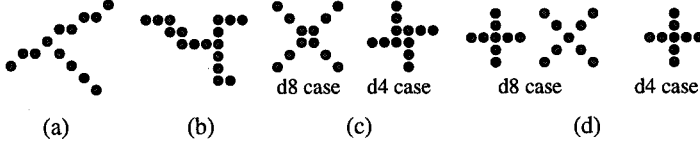


Fig. 2.1: (a) An 8-connected skeleton. (b) An 4-connected skeleton. (c) Some pathological cases that can be solved by special masking. (d) Admissible maximal cases after good skeletonization.

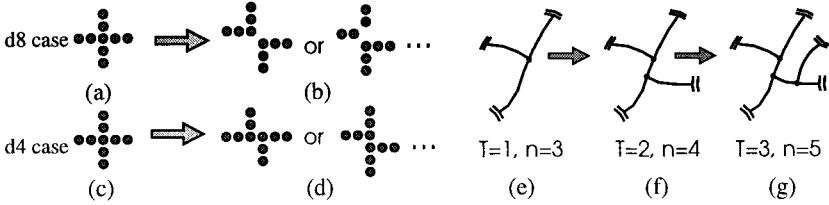


Fig. 2.2: (a-d) Other possible admissible configurations. (e-g) To append a branch to a crossing adds a new TEP.

Notice further that configurations as the ones shown in Figure 2.1 (d) may appear also (and with a great probability) as shown in Figure 2.2 (b) and (d). So, we note the following:

Lemma 1. [2] For (Z^2, d_k) with $d_k \in \{d_4, d_8\}$, any crossing point should be decomposed into a set of three-edge-points (TEPs).

Proof: It suffices to find a case where a crossing is generated by a set of TEPs to show that this is the general way to decompose a crossing point into TEPs in Z^2 .

In general, any crossing (in the real case) will not be found in its natural form. Instead, it will appear decomposed into crossings of less order as shown in Figure 2.2 (b) and (d). Thus, the general decomposition of any crossing is into TEPs.

As a result we have the following

Theorem 1. [2] A crossing point with n branches is equivalent to $n - 2$ TEPs. i.e. $TEPs = n - 2$.

Proof. By construction (see Figure 2.2 (e-g)), consider the case of the minimal crossing, i.e. a crossing with just one TEP (Figure 2.2 (e)) then $n - TEPs = 2$. If we append a new branch to this crossing point, a new TEP is added (Figure 2.2 (f)), and $n - TEPs = 2$. It is not difficult to see that if we continue appending

new branches (Figure 2.2 (f)), $n - \text{TEPs} = 2$ remain the same, then $\text{TEPs} = n - 2$.

3. Feature Extraction and Object Description

In this section the feature extraction process is described. This process is needed to obtain the reference model for each object to be further recognized. Lots of features have been used in the past to derive the model of an object. These include computing Fourier shape descriptors [14] and [7], different types of moments [4], [3] and [6], as well as classical shape measures of compactness, elongatedness, and son on [8].

The kind of features used in this work are topological in nature. They have been chosen firstly due to the representation chosen to describe the structure of the shape, and secondly, because topological features have proven to be very nice global descriptors in shape recognition [12]. The two topological features chosen here are the *terminal points* (TPs), i.e. points with just one neighbor and the *three-edge-points* (TEPs), i.e. points with only three neighbors. These features have been already used in the computation of the Euler Number of a binary shape [2]. The extraction of these features is as follows. From a binary image containing one or more binary objects:

1. **Skeletonize** the image until a 4-skeleton for each of object is obtained. The method used in this case (see [1]) profits the advantages of the *Fast Distance Transformation* (FDT) to rapidly obtain the desired skeleton. For this, it uses a logical operator over the FDT instead of the classical morphologic operators over the discrete array (erosion and dilation), to obtain a much faster algorithm.
2. **Label** each pixel as:
 - a **terminal point** if the number of its 4-neighbors equals one.
 - a **crossing point** if the number of its 4-neighbors is greater or equal to three.
3. **Decompose** each crossing point p into its number of TEPs. This number, as we said and as it was shown in [2], equals the number of branches composing the crossing point - 2, or equivalently, $N_4(p) - 2$.

Once each skeleton has been decomposed into its TPs and TEPs, their number is used to describe the object. An object is then described by a label (the label identifying the object) and two numbers: the number of TPs and the number of TEPs of its associated skeleton. This information is used to build the database of models. It is a list of descriptions ordered by magnitude of the two elements: number of TPs and number of TEPs. Figure 3.1 (a) shows some shapes and their corresponding descriptions. Figure 3.1 (b) shows their positions along the database.

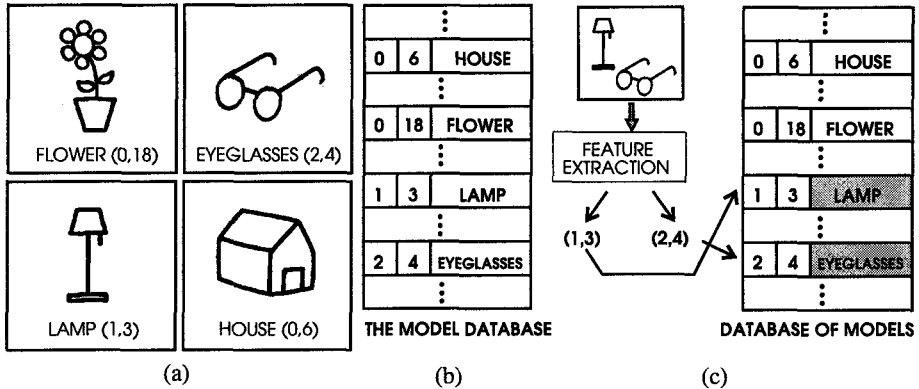


Fig. 3.1: (a) Some objects and their descriptors. (b) Positions of these models in the database. (c) The searching process.

4. Object Identification

Once the database of models has been constructed, a binary image containing one or more objects is presented to the system. Each object is then preprocessed as explained in section 3 to obtain respectively: its skeleton and its numbers of TPs and TEPs. The number of PTs and TEPs is then used as an access key to retrieve from the database of models the corresponding models. This process is depicted in Figure 3.1 (c).

5. Results and Discussion

The methodology proposed in sections 3 and 4 is applied in the recognition of flat objects comprising both rigid and articulated instances. The reference patterns used to train the system are shown in Figure 5.1 (a). Their corresponding skeletons along with their descriptors are shown in Figure 5.1 (b).

5.1. Rigid Case

The patterns used to test the system's performance in this first case are shown in Figure 5.1 (c). It is not difficult to see that they are copies of the same objects shown in Figure 5.1 (a) but subjected to a combination of a translation, a rotation and a change of scale. Note that in all cases (as expected) all objects were successfully recognized.

5.2. Flexible Case

A more interesting problem to tackle with, is the case of recognizing articulated objects. i.e. objects with a certain number of rigid parts and links maybe

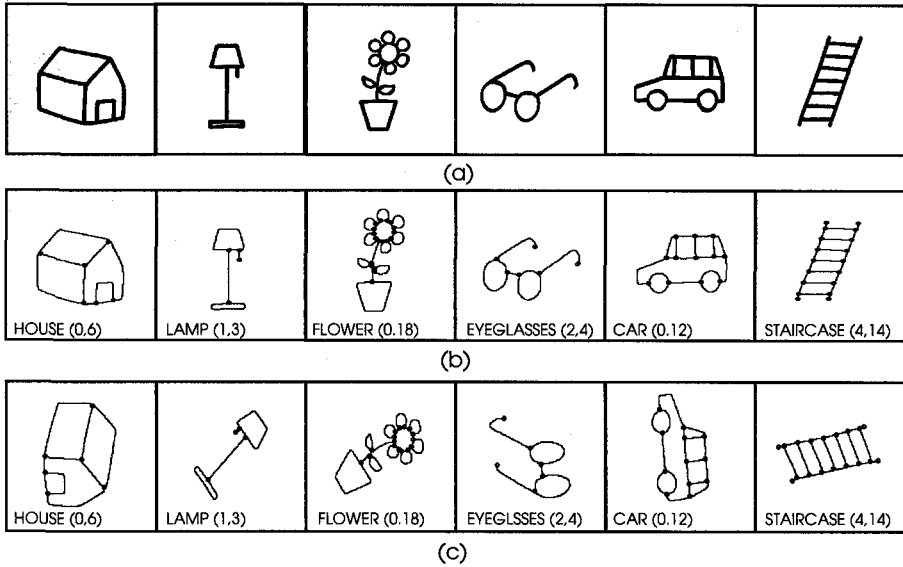


Fig. 5.1: (a) Reference patterns. (b) Their skeletons and descriptors. (c) Test patterns.

different joining them. Two different classes of objects were used in this case. The first class comprises flexible objects composed of two or at most three rigid parts with a rotating or a translating axis: scissors, verniers, pliers, and so on; the second class comprises objects with a great number of rigid parts and different kind of links joining them: worms, spiders, crabs, and so on. The patterns used to test the system belonging to the first class are shown in Figure 5.2 (a). The matching results are shown in Figure 5.2 (b). Note again that in all cases, all instances were well recognized.

In the second phase of the experiment, a couple of sequences of images representing the movement of an object among other objects in the scene, were used to test the system. Both sequences are shown in Figures 5.3 (a) and 5.4 (a). The goal here was to follow the movement of a given object among the others given its modeling parameters. This was carried out by marking the object's pixels with a different symbol. In this case, the object's pixels appear darker than the rest. The tracking results are shown in respectively in Figures 5.3 (b) and 5.4 (b). Note again the effectiveness of the method in both cases.

5.3. Discussion

It is obvious that in both cases the method will present some problems. For example, it suffices that any of the appendices of the staircase shown in Figure 5.1 (b) and (c) disappears so that the system fails. Similarly, it suffices that the

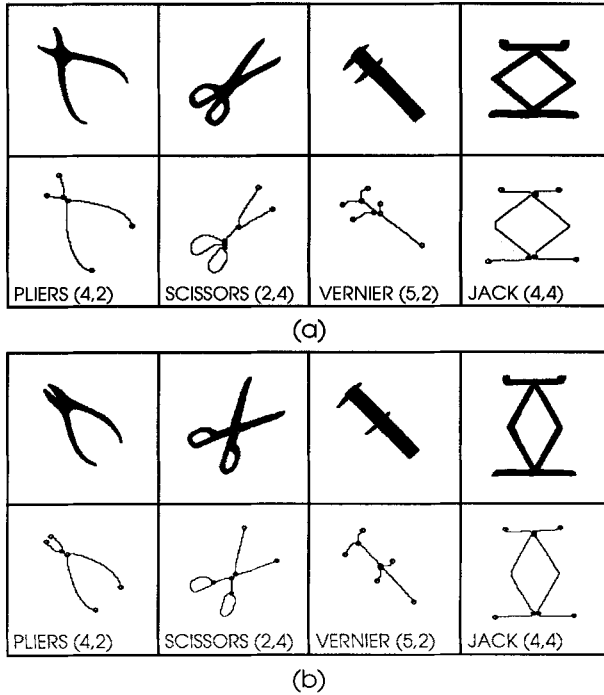


Fig. 5.2: (a) Flexible reference patterns. (b) Flexible test patterns.

worm of Figure 5.4 cross a couple of its multiple legs, so the same occurs. The same problem will appear if the skeleton's connectivity is lost or if some parasite branch appears at the moment of the obtaining of the skeleton.

It is also clear that the exclusive use of the TPs and the TEPs for the modeling of wire-shapes cause that a lot of objects with different shape have the same number of TPs and TEPs. This is the case, for example, of the eyeglasses of Figure 5.1 and the scissors of Figure 5.2. In this case the system will give as a result a list of those objects having the same pair (TPs, TEPs).

6. Conclusions and Future Work

In this paper, a simple method for the recognition of two-dimensional wire-objects was presented. It is based on a topological model of the object (its skeleton) using as features the number of terminal points (TPs) and the number of three-edge-points (TEPs), i.e. points with only three neighbors in the objects' skeleton. Both the rigid and the articulated cases were tackled with nice results.

The principal merits of the proposed approach are its originality, execution speed and robustness in the presence of geometrical changes.

One disadvantage of the proposed method is that it completely topologically

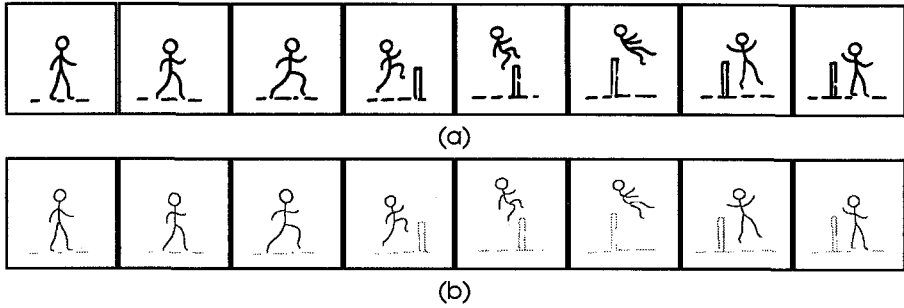


Fig. 5.3: (a) Test sequence. (b) Tracking the man-pattern.

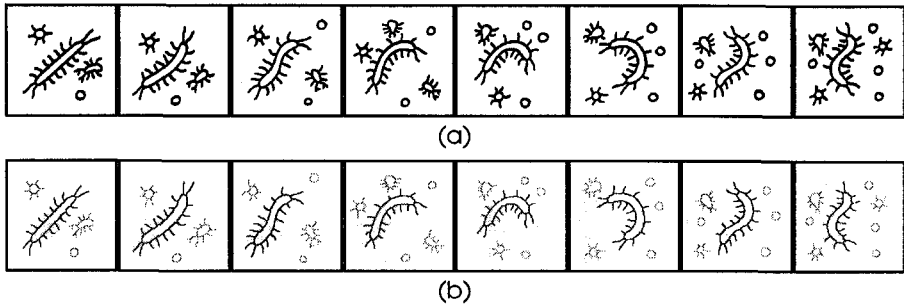


Fig. 5.4: (a) Test sequence. (b) Tracking the worm-pattern.

dependent. This means that the object's topology cannot be broken. If this is the case, a different method should be used. A local one could be a good option.

Another characteristic of the present approach is that there are a multitude of shapes sharing the same set of TPs and TEPs. This problem could be partially solved for example by using (along with the number of TPs and TEPs) their distribution along the skeleton to obtain a more complete model, by means of a graph for example.

As a future work, we are planing to tackle the more challenging problem of recognizing 2-D rigid and non-rigid objects when the object is occluded by other objects or by itself, when some of the objects' information is missing and when some noise is present in the image. Figures 6.1 (a), (b), (c) and (d) show these four cases. Although it is true that the recognition of flat objects from their skeletons is still lacking a real success, perhaps if combined with other techniques like contour coding, special scanning sequences, model matching, etc. it may turn to be a winner.

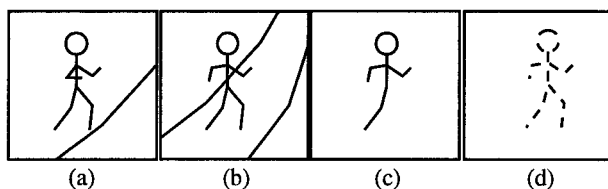


Fig. 6.1: (a) An object occluding itself. (b) Mutually occluding objects. (c) Incomplete object. (d) Noisy object.

References

- [1] J. L. Díaz-de-León S. *Skeletonization algorithms for binary images*. Master's thesis, CINVESTAV - IPN, Depto. de Ing. Eléctrica, Secc. de Control Automático, August 1993. In Spanish.
- [2] 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):471–476, 1996.
- [3] J. Boyce and W. Hossack. Moment invariants for pattern recognition. *Pattern Recognition Letters*, 1:451–456, 1983.
- [4] M. Hu. Visual pattern recognition by moment invariant. *IRE Transaction on Information Theory*, 12:179–187, 1962.
- [5] T. Knoll and R. Jain. Recognizing partially visible objects using feature indexed hypothesesoccluded parts. *IEEE J. of Robotics and Automation*, 2(1):3–13, 1986.
- [6] Z. Mingfa nad S. Hasani, S. Bhattarai, and H. Sing. Pattren recognition with moment invariants on a machine vision system. *Pattern Recognition Letters*, 9:175–180, 1989.
- [7] E. Persoon and K. S. Fu. Shape discrimination using fourier descriptors. *IEEE Trans. Syst. Man, Cybern.*, 7:170–179, 1977.
- [8] A. Rosenfeld and A. Kak. *Digital Picture Processing*. Academic Press, New York, 1976.
- [9] D. Rutovitz. Pattern recognition. *J. Royal Statist. Soc.*, 129:504–530, 1966.
- [10] H. Tamura. A comparison of line thinning algorithms from a digital geometry viewpoint. In *4th. Int. Conf. on Patt. Recog.*, pages 715–719, 1978.
- [11] J. Turney, T. Mudge, and R. Voltz. Recognizing partially occluded parts. *IEEE Trans. Patt. Anal. and Mach. Intell.*, 7:410–421, 1985.
- [12] H. Yang and S. Sengupta. Intelligent shape recognition for complex industrial tasks. *IEEE Control Systems Magazine*, 23–29, June 1988.
- [13] S. Yokoi, J. Toriwaki, and T. Fukumura. An analysis of topological properties in digitized binary pictures. *CGIP*, 4:63–73, 1975.
- [14] C. Zhan and R. Roskies. Fourier descriptors for plane closed curves. *IEEE Trans-action on Computers*, 21:269–280, 1972.