An algorithm for finding closed curves

Ming Xie and Monique Thonnat

INRIA Sophia-Antipolis, 06565 Valbonne, France

Received 31 May 1991
Revised 19 September 1991

Abstract


Segmentation is a basic operation in pattern recognition. Without a previous separation of objects in an image, shape analysis of imagery pattern cannot be carried out easily. In this paper, we will present an algorithm allowing the segmentation of contour maps into closed curves. The basic idea consists of first establishing the neighbourhood relationship of contour chains and then of finding closed curves based on a grouping technique using a 'minimal angle' criterion.

Keywords. Pattern recognition, segmentation, contour chains, closed curves.

1. Introduction

Shape analysis plays an important part in pattern recognition. Region based descriptions and boundary based descriptions of shapes are two alternative approaches largely investigated in the past. In [1], a survey of work in this domain before 1978 is published. For the boundary based descriptions, the research primarily focuses on the so called 'fitting techniques'. In the literature, we can find three categories of such methods: (a) the line or curve fitting techniques [2,3,5], (b) the polygonal fitting techniques [4,9] and (c) the circular or elliptical fitting techniques [4,11,12,13]. For the last two categories, the methods proposed require a basic assumption about the segmentation of contour maps, i.e., the closed curves should be previously extracted from contour maps so that the fitting techniques could be applied. This assumption reveals the importance of the segmentation task. Before any shape analysis can be carried out, the first operation needed is the separation of the patterns to be recognized in images. In general, the shape of a visual object takes a variety of forms. But the closed ones give rise to more significant information about shapes. Thus the segmentation of contour maps into closed curves becomes a meaningful task.

In this paper, we will present an algorithm allowing the segmentation of contour maps into closed curves. The input of our algorithm is a set of contour chains obtained by an edge linking operator (see [10]). We characterize each chain by its two endpoints (head and tail). Thus a closed curve is defined to be a subset of chains with their endpoints connected. We first try to identify the list of neighbouring chains connected to each endpoint. Such a neighbourhood relationship classically leads to build a connection graph of chains with the chains as the arcs and the endpoints as the nodes, and then to find the closed paths of the graph. The algorithm described in [7] explored this idea in order to find the metachains (including closed curves). In practice, such strategy is computationally expensive because one must take into account all the possible combinations of chains (see [6]). To solve this problem, we shall propose here a grouping technique based on a 'minimal
angle' criterion for finding the subsets of chains that form closed curves. The two basic operations of our algorithm are (a) the identification of neighbouring chains for each chain and (b) the grouping of chains in order to find the closed curves.

2. Identifying the neighbouring chains

We have used the edge linking operator described in [10] to obtain the set of contour chains in a contour map. The contour chains produced by this operator are single curve segments, that means no chain will have any branches. For example, the contour map shown in Figure 1 is composed of fifteen chains (C1–C15) according to the edge linking operator used.

Consider a two-dimensional coordinate system OXY in the image plane and let \{Ci, i = 1, ..., n\} be a set of n contour chains in this 2D space. Since each contour chain corresponds to a single curve segment, it can be parameterized by its two endpoints. This leads to \(C_i = (H_i, T_i) (i = 1, ..., n)\), where \(H_i\) and \(T_i\) denote the head and the tail of \(C_i\), respectively. The designation of head point or tail point for an endpoint is rather arbitrary. But each chain will have only one head point and one tail point. In this way, the neighbourhood relationship between any two chains can be defined as follows:

**Definition 1.** For any two endpoints \(P_i\) and \(P_j\) (\(P_i\) and \(P_j\) may be either head points or tail points), if the Euclidean distance \(||P_iP_j||\) verifies the following relation:

\[ ||P_iP_j|| \leq S_d \tag{1} \]

where \(S_d\) is a positive threshold, then we call \(P_i\) and \(P_j\) two connected points.

**Definition 2.** Given a chain \(C_i\), if another chain \(C_j\) has one endpoint connected to the head point of \(C_i\), then we call \(C_j\) a head neighbour of \(C_i\). Otherwise, if \(C_j\) has one endpoint connected to the tail point of \(C_i\), then we call \(C_j\) a tail neighbour of \(C_i\).

According to the above definitions, one chain may have two sets of neighbours: a set of head neighbours with respect to its head point and a set of tail neighbours with respect to its tail point. For the example in Figure 1, the chain \(C_5\) may have \(\{C_1, C_2, C_6\}\) as its head neighbours and \(\{C_4, C_9, C_{10}\}\) as its tail neighbours. So, for a set of chains to form a closed curve, a necessary condition will be that all these chains should not have any empty set of neighbours, that is:

**Necessary condition.** If a contour chain belongs to a closed curve, then its two sets of neighbouring chains will not be empty.

From this necessary condition, we can eliminate all the non-connected or single-connected chains which have at least one empty set of neighbouring chains. Therefore, the identification of neighbouring chains can be carried out by the following three steps:

**Step 1.** For each endpoint of chains, build the list of its connected points according to the relation (1).

**Step 2.** Knowing all the connected points, for each contour chain, build respectively the list of its head neighbours and the list of its tail neighbours.

**Step 3.** Eliminate all the chains which have at least one empty set of neighbouring chains.

For the example shown in Figure 1, after the identification of neighbouring chains, only the chains \(\{C_5, C_6, C_7, C_{10}, C_{12}, C_{13}\}\) will be kept for further treatment of finding closed curves.
3. Finding closed curves using a minimal angle criterion

Once the neighbours of each chain have been identified, the next problem will be how to find the subset of chains forming a closed curve, knowing the neighbourhood relationship of the chains. To solve this problem, one simple solution consists of tracing the connected chains by starting from a selected chain. If the tracing operation returns to the starting chain, we can say that all the chains passed form a closed curve. For the example in Figure 1, if we select the chain C5 as the starting chain and if we succeed to follow the chains \{C_{10}, C_7, C_6\} before reaching again to the chain C5, then we have found a closed curve composed of the chains \{C_5, C_{10}, C_7, C_6\}.

Now we can see that the main difficulty of this kind of grouping technique is how to decide which one among the set of neighbouring chains is to be traced in the next tracing step. In our case, we will use a decision strategy based on a 'minimal angle' criterion. The basic idea of this decision strategy can be explained as follows:

Let \(C_i\) be the current chain traced and we want to decide which chain in the set of its \(m\) tail neighbours \(\{C_j, j = 1, \ldots, m\}\) should be selected as the next chain to be traced. Without loss of generality, we assume that all the tail neighbours have their head points connected to the tail point of the chain \(C_i\) (remember that the designation of head point or tail point is arbitrary). Moreover, suppose that a reference point \(P_r\), around which we expect a closed curve to be found, is known. For example, if we want to find a closed curve enclosing the area on the right-hand side of a chain \(C_i\), then we choose a point within that enclosed area as the reference point. We shall discuss how to determine a reference point later on.

Given the tail point \(T_i\) of the chain \(C_i\) and the reference point \(P_r\), we can define a unit reference vector \(V_r\) which starts at \(T_i\) and points to \(P_r\), that is:

\[
V_r = \frac{T_i P_r}{\|T_i P_r\|}. \tag{2}
\]

For each tail neighbour \(C_j\), we can calculate its unit tangent vector \(V_j\) which starts at the head point \(T_j\) and points back towards the chain. Similarly, we can also calculate the unit tangent vector \(V_r\) at the tail point \(T_i\) of the chain \(C_i\).

Now, the intersection angle between the vector \(V_j\) and the vector \(V_r\) can be defined as:

\[
\theta_j = \arccos(V_j \cdot V_r), \quad j = 1, \ldots, m, \quad \tag{3}
\]

where \(\cdot\) means the inner product of two vectors. Since the \(\arccos(\ )\) function varies from 0 degree to 180 degree, any two vectors symmetrical to the vector \(V_r\) will have the same intersection angle. In order to be able to distinguish if a vector is to the right or the left of the reference vector \(V_r\), we propose to calculate the intersection angle by the following equations:

1. If the vector \(V_r\) is at the right side of the tangent vector \(V_j\), then compute \(\theta_j\) \((j = 1, \ldots, m)\) via:

\[
\begin{align*}
\theta_j &= \arccos(V_j \cdot V_r), \\
&= 2\pi - \arccos(V_j \cdot V_r),
\end{align*}
\]

2. If the vector \(V_r\) is at the left side of the tangent vector \(V_j\), then compute \(\theta_j\) \((j = 1, \ldots, m)\) via:

\[
\begin{align*}
\theta_j &= 2\pi - \arccos(V_j \cdot V_r), \\
&= \arccos(V_j \cdot V_r),
\end{align*}
\]

In this way, the value of an intersection angle varies from 0 degree to 360 degree. To determine if a vector is to the right or the left of another vector, we just need to check the sign of the non-zero component of the cross product of these two vectors.

Knowing the intersection angles associated with the neighbouring chains, the decision of the next chain to be traced can be made by the following 'minimal angle' criterion, that is:

\[
C_k, \quad k \in (1, m) \quad \text{such that} \quad \theta_k = \min_{j=1, \ldots, m} \{\theta_j\}. \tag{6}
\]

This criterion means the chain neighbour \(C_k\) will be selected as the next chain to be traced if the intersection angle \(\theta_k\) has the minimal value among
Figure 2. The ‘minimal angle’ criterion based decision strategy. The chain $C_i$ has four neighboring chains $\{C_1, C_2, C_3, C_4\}$. Since the reference point $P_r$ is at the right side of the tangent vector $V_r$, the next chain to be traced will be $C_1$ since its tangent vector $V_1$ has the minimal intersection angle with $V_r$.

For the example shown in Figure 2, as the reference vector $V_r$ is at the right side of the tangent vector $V_1$, thus the tangent vector $V_1$ has the minimal intersection angle with the vector $V_r$ and the chain neighbour $C_1$ will be selected as the next chain to be traced.

Remember that the presented tracing process based on the ‘minimal angle’ criterion depends on the position of reference point $P_r$. An open question will then be how to determine a reference point for each tracing process. As mentioned above, a tracing process begins at a starting chain; a reference point is a point around which we expect a closed curve to be found. Therefore, two solutions will be suitable for determining a reference point. The first solution consists of trying two reference points: one at the left side of the starting chain and the other at the right side of the starting chain. This solution implies that a tracing process will be performed two times for each starting chain. The second solution consists of using the common centre of gravity of all the chains already traced. Each time a new chain has been traced, we update the reference point by computing the new common centre of gravity of the traced chains. Due to the convexity of a closed curve, the constantly updated reference point will stay inside the closed curve that we seek. In our implementation of the algorithm, we have adopted the second solution.

Now we summarize the algorithm of finding closed curves (which is a recursive procedure) as follows:

**Step 1.** If all the chains have been treated, stop the searching process. Otherwise, choose arbitrarily one chain as starting chain among the chains not belonging to any closed curve (initially, no chain belongs to a closed curve). Compute the centre of gravity of the starting chain and let it be the reference point.

**Step 2.** For the starting point, let one of its two endpoints be the initial control point and the other be the termination point. The control point is used to identify a set of neighboring chains (Definition 2) from which we shall select the next chain to be traced. But the termination point is used to determine the end of a tracing operation, that is to determine whether the tracing operation has succeeded in reaching this point. Either one of the two endpoints of the starting chain may be determined to be the termination point. This just changes the tracing direction but will not affect the tracing outcome.

**Step 3.** According to the ‘minimal angle’ criterion, decide upon the next chain to be traced among the neighboring chains relative to the control point. This selected chain is called the current traced chain. Then compute the common centre of gravity of the already traced chains and update the reference point.

**Step 4.** Of the two endpoints of the current traced chain, we identify the one which is not connected to the control point. We then let this endpoint be the new control point. This updating of control point means that at the next step the chain to be traced will be selected from the neighboring chains relative to this new control point. In such a way, the tracing operation will go on.

**Step 5.** If the new control point is a point connected to the termination point, a closed curve is found; go to Step 1. Otherwise, continue the current tracing operation with Step 3.

4. Discussions

The significance of the ‘minimal angle’ criterion can be understood by the fact that it tends to find
the closed curves which will have the minimal enclosed areas. For the example shown in Figure 1, it is easy to see that the 'minimal angle' criterion will lead to find the two closed curves \( \{C_5, C_6, C_7, C_{10}\} \) and \( \{C_7, C_8, C_{13}, C_{12}\} \). But we shall fail to find the closed curve \( \{C_5, C_6, C_8, C_{13}, C_{12}, C_{10}\} \). Depending on the application contexts, if such kind of closed curves (called 'outer closed curves') ought to be recognized, it may be possible to use an alternative 'maximal angle' criterion. In this paper, we shall not deal with the case of outer closed curves.

However, if we name the closed curves found by our algorithm the elementary closed curves (with minimal enclosed areas), then one can easily observe that any non-elementary closed curve will be the union of a subset of connected elementary closed curves with their common chains eliminated. For the above example, if we eliminate the common chain \( C_7 \) of the two connected closed curves \( \{C_5, C_6, C_7, C_{10}\} \) and \( \{C_7, C_8, C_{13}, C_{12}\} \), then the union of these two elementary closed curves produces a non-elementary closed curve \( \{C_5, C_6, C_8, C_{13}, C_{12}, C_{10}\} \). From such consideration, we can state that finding the elementary closed curves is a fundamental operation and is useful for analysing more complicated closed curves.

5. Experimental results

Our algorithm has been implemented in C++ on a Sun4 workstation. We have used the algorithm presented in [8] for edge extraction and the algorithm presented in [10] for edge linking.

Example 1. Figure 3 shows the results obtained with a real image of an indoor scene. Fifty-eight closed curves have been found from the set of 339 contour chains shown in Figure 3c.

Example 2. Figure 4 shows the results obtained with a real image of an outdoor scene. Fifty-two closed curves have been found from the set of 698 contour chains shown in Figure 4c.

Example 3. Figure 5 shows the results obtained with a real image of another outdoor scene.

Eighty-four closed curves have been found from the set of 905 contour chains shown in Figure 5c.

Table 1 displays the data concerning these experimentations.

In Example 1, we fail to find the large closed curve around the three circular closed curves. In Example 2, we fail to find the closed curves corresponding to the windows of the building in the background. These failures are due to the fact that we do not take into account the neighbourhood relationship which may be defined by the connection relationship between an endpoint and the body of a chain. To overcome this, a further investigation is necessary.

In our algorithm, the only working parameter is the threshold \( S_d \) used to control the neighbourhood relationship of the endpoints of chains (see Definition 1). The adjustment of this parameter depends on the quality of contour maps. If relatively great gaps are present in edges and at the junctions of chains, the threshold \( S_d \) must be set to a relatively great value. In our experimentations with a variety of real images (single images or sequences of images), we have chosen \( S_d \) to be 3 or 4 pixels. The obtained results are stable.

The processing time needed by our algorithm obviously depends both on the number of chains in the contour map and on the number of closed curves finally found. However, the processing time may be influenced by the value of \( S_d \) since a higher value of \( S_d \) will generally lead to a higher number of neighbouring chains for a chain. Moreover, a great value of \( S_d \) may also lead to find wrong closed curves. For example, a 'U' type curve will be considered as a closed curve if \( S_d \) is greater than the distance between the two endpoints of this 'U' type curve. Under the context of our experimentation (using the algorithm in [8] for

<table>
<thead>
<tr>
<th>Example</th>
<th>Number of chains</th>
<th>Number of closed curves</th>
<th>Threshold ( S_d )</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example 1</td>
<td>339</td>
<td>58</td>
<td>4 (pixels)</td>
<td>0.32 (s)</td>
</tr>
<tr>
<td>Example 2</td>
<td>698</td>
<td>52</td>
<td>3 (pixels)</td>
<td>0.78 (s)</td>
</tr>
<tr>
<td>Example 3</td>
<td>905</td>
<td>84</td>
<td>3 (pixels)</td>
<td>0.92 (s)</td>
</tr>
</tbody>
</table>
edge extraction and the algorithm in [10] for edge linking), choosing a value for $S_d$ greater than 3 or 4 pixels is not recommended.

From the results shown here, we can see that some closed curves correspond to closed planar curves in 3D space; some correspond to the windows of a car or building; some correspond to numbers printed in the image; some correspond to the landmarks of roads. To find such closed curves, our algorithm does not need a priori knowledge about the shape description of the closed curves to be found (unlike Hough Transform based techniques (see [2,3])). This is an advantage of our algorithm if such a shape description is not feasible.

Figure 3. Example 1: (a) a 512 x 512 real image of an indoor scene; (b) the contour map; (c) the set of contour chains which were the input to our algorithm; (d) the closed curves found by our algorithm.
Our work was supported by the Eureka project PROMETHEUS. The results of our algorithm will contribute to the study of the classification of road obstacles, the study of the detection of traffic signs and the investigation of direct 3D reconstruction of closed curves in the 3D space of the camera mounted on a car. Of course, the results of our algorithm may constitute input for fitting algorithms such as in [12] and [13].

6. Conclusions

We have presented here a segmentation algorithm that allows the identification of closed curves from
a set of contour chains. The basic idea consists of establishing first the neighbourhood relationships of the contour chains and then of finding the closed curves by a tracing process which is based on the use of a 'minimal angle' criterion. We think that the 'minimal angle' criterion ensures a fast algorithm, since we do not need to explore all the possible combinations of chains at each closed-curve tracing process. Another interesting aspect of our algorithm is that we can find closed curves which can either be modeled mathematically or not. So, our method is more general than Hough Transform based techniques [2,3] since the latter are just suitable for finding closed curves (e.g.,
circles or ellipses) which can be described mathematically.

We are of the opinion that the segmentation of the contour chains into closed curves is a fundamental and meaningful task in pattern recognition. The success of this operation is a necessary step before curve fitting techniques can be applied to complex real images. It must be pointed out that the algorithm presented uses only the neighbourhood relationship of the contour chains defined by the connection relationship of the endpoints of chains. However, the case where the neighbourhood relationship may be defined by the connection relationship between an endpoint and the whole body of a chain is not considered. This may be object of future study. On the other hand, we think that the principle of our algorithm is applicable to the problem of finding the polygons (closed line segments) from a set of line segments.

Acknowledgements

The authors would like to thank the anonymous referee for his useful comments and the Eureka project PROMETHEUS for its financial support.

References