COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING 44, 119–138 (1988)

A Topology-Based Component Extractor for Understanding Electronic Circuit Diagrams*

CHIN-SHYURNG FAHN, JHING-FA WANG, AND JAU-YIEN LEE

Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan 70101, Republic of China

Received August 6, 1986; revised April 27, 1988

An automatic understanding system using the techniques of image processing, pattern recognition, and artificial intelligence has been developed for electronic circuit diagrams. Part of the system is presented to extract three categories of essential components: circuit symbols, characters, and connection lines. Each essential component consists of a set of picture segments which are appropriately detected by a segment tracking algorithm. A heuristic piecewise linear approximation algorithm is proposed to approximate picture segments for primitive recognition. On the basis of topological context, a one-pass manner called the relational best search method applies a depth first search technique uniting a set of specified rules during the traversal of a circuit diagram. This method combines the constituents of each circuit symbol or character into a cluster. All the clusters together with the remaining components are extracted and grouped into the three categories as soon as the traversal is finished. A variety of electronic circuit diagrams have been used for testing the component extractor. So far, the present extractor has shown favorable results.

I. INTRODUCTION

In pace with the increasing complexity and diversity of electronic circuits, the use of computer-aided design (CAD) tools has been becoming more and more desirable for circuit design to augment productivity. For an integrated CAD system of electronic circuits, the first problem encountered is how to input the initial design data concerning circuit functions into a computer promptly and precisely. A two-dimensional schematic drawing provides an efficient manner of representing the logical structure of a circuit. Conventionally, the schematic drawing is entered into the CAD system using interactive methods with light pens or data tablets [1–4]. These hand-digitizing approaches, however, depend mainly on human operators whose performance is generally inadequate and variable owing to the inconvenience of the awkward input process. During this manual operation, erroneous data are apt to be mixed with correct data. As a result, the design cycle is normally prolonged and the design cost grows expensive, especially for a large-scale circuit. An automatic understanding system for electronic circuit diagrams can directly capture the data from real drawings or check on the consistency of the input data. Alleviating the time-consuming and error-prone bottlenecks in the hand-digitizing approach, the system is obviously superior to the human operators.

A complete electronic circuit diagram usually contains three categories of essential components: circuit symbols, characters, and connection lines. In the last few years, there have been several approaches based on the techniques of image processing and pattern recognition to make machines recognize circuit diagrams

*This research was supported by the National Science Council of the Republic of China under Grant NSC76-0404-E006-09.

0734-189X/88 $3.00

[5–13]. Although text annotations are indispensable information for characterizing a circuit, most of the proposed approaches neglect the characters in an actual drawing and thus offer no definite algorithms to process the lettering on the drawing. In [6, 7], the identification of characters is eliminated entirely. Moreover, some of them impose too severe drawing conventions on designers to sketch circuit diagrams. For example, the characters in [5] must be isolated components of a prescribed size, and the slant lines in [5–8, 11] are not permitted to constitute the connection lines among circuit symbols. Still worse, some circuit symbols in [6, 8, 9] must be expressed by odd patterns. This practice makes the altered symbols lack the general recognition of communication in the electronic engineering discipline.

An automatic understanding system described here has been developed for comprehending electronic circuit diagrams in printed manuals or freehand drawings of a fair quality. The present system can deal with text annotations as well as logical structures under only a few constraints. All circuit symbols can be roughly laid in the four regular orientations (i.e., right, up, left, and down), but characters may not be upside down. Usually, the size of a character is relatively smaller than that of a circuit symbol. Figure 1 shows the principal exemplary models of the circuit symbols handled in the present system. The exemplary models of alphanumerical characters are well known but not shown here. For a connection line, all its path is not necessarily either horizontal or vertical. In addition, broken lines and characters

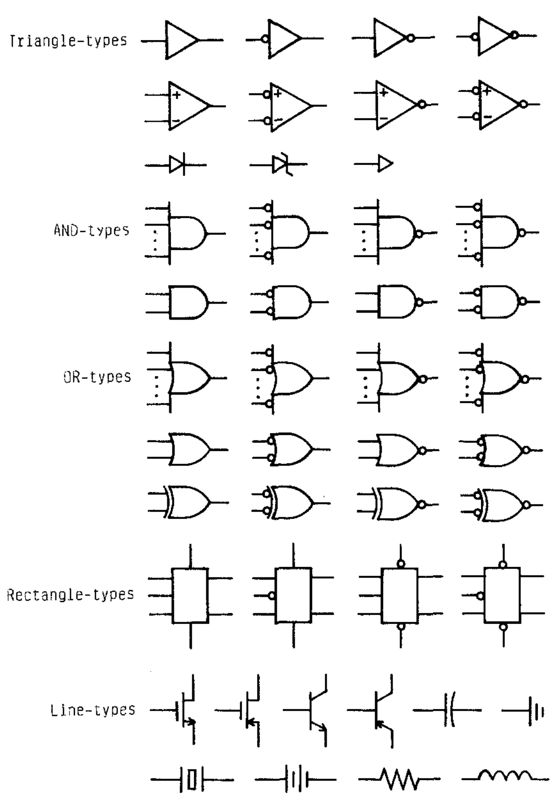


FIG. 1. Principal exemplary models of circuit symbols.

connected with other essential components are also allowed. For practical use, the mergence of the partitions of a large-scale circuit diagram can be manipulated by this system. The hierarchical process of understanding circuit diagrams includes a series of preprocessing, extracting, synthesizing, recognizing, and interpreting phases. In this paper, we concentrate on the principles of the extracting phase to obtain the essential components from electronic circuit diagrams.

II. PRIMARY STRATEGY

The component extractor traverses a circuit diagram by following the skeletons of objects. Each object in a circuit diagram is composed of a set of picture segments which are delimited with feature points and properly tracked in the course of the traversal. By means of piecewise linear approximation, each picture segment is then described as a sequence of vectors for primitive recognition. During the traversal of the circuit diagram, the constituents of each circuit symbol or character are combined into a cluster by analyzing the context of picture segments within certain confines. On the basis of the qualities embedded in the exemplary models of essential components, each cluster of related primitives is identified as a circuit symbol or character. If the picture segments are excluded from the clusters, they are assigned to the constituents of connection lines.

In this method, a depth first search technique uniting a set of specified rules called the relational best search (RBS) has been devised to perform the extraction of essential components once and for all. It is a great contrast to the conventional component extraction methods that make redundant examinations of a pile of fragmentary primitives [9, 10] or require many passes to verify the primitives piece by piece [7]. In consequence, not only can the extracting time of the RBS method be greatly shortened, but also the extracting precision can be enormously improved. Additionally, the RBS technique traversing the entire image of a circuit diagram is contrary to the split-and-merge methods that may lose track of circuit information owing to the mismatch on the boundaries of predetermined meshes [5, 6, 8, 9, 11, 13]. For these reasons, the processing efficiency of our component extractor is better than that of the traditional ones.

III. SEGMENT TRACKING

After running preprocessing operations, each object in a binary image is thinned down to a skeleton of a unit thickness.

A. *Picture Segment*

Let the binary image B be mapped into a matrix $[b_{ij}]_{cxd}$, where each pixel b_{ij} has coordinates (i, j) normalized by the unit length of a Cartesian coordinate system.

DEFINITION 3.1. The *color function* of a pixel b is defined as

$$C(b) = \begin{cases} 1, 2, 3, \text{or } 4 & b \text{ is an object pixel labelled or not,} \\ 0 & b \text{ is a background pixel or a deleted object pixel.} \end{cases}$$

DEFINITION 3.2. A *path* from p_0 to p_n in B is a sequence of connected pixels such that $\prod_{i=0}^{n} C(p_i) \neq 0$; for a closed path, $p_0 = p_n$, and for an open path, $p_0 \neq p_n$.

n_3	n_2	n_1
n_4	b	n_0
n_5	n_6	n_7

FIG. 2. Designations of the eight neighbors of a pixel b in a 3-by-3 window.

DEFINITION 3.3. The *path length* denoted as $L(A)$ is measured by counting the number of pixels in a path A.

Each object is a collection of drawing lines whose constitutive pixels have nonzero color values. In the skeleton of the object, some pixels are characterized as feature points when they form the terminal points or the joint points of the drawing lines. In the Cartesian coordinate system, each pixel b has eight neighbors as are designated in Fig. 2.

DEFINITION 3.4. The *neighboring function* of an object pixel p with the color value $C(p) = 1$ is defined as

$$N(p) = \sum_{i=0}^{7} \Delta n_i, \text{ where } \Delta n_i = \begin{cases} 1 & C(n_i) \neq 0 \text{ and } C(n_{i+1}) = 0, \\ 0 & \text{otherwise,} \end{cases}$$

and $n_8 = n_0$.

The neighboring value is used for describing the point p. If $N(p) = 0$, then p is an isolated point. If $N(p) = 2$, then p is a chain point. In other cases, p is a *feature point* defined as:

(a) If $N(p) = 1$, then p is an *end point*;
(b) If $N(p) = 3$, then p is a *branch point*;
(c) If $N(p) = 4$, then p is a *cross point*.

Figure 3 indicates the typical types of a point p with its neighboring points. Among these types, all the feature points are regarded as the dividing points that perspicuously separate a circuit diagram into a set of picture segments during the extraction of essential components.

DEFINITION 3.5. A *picture segment* consists of a path $p_0 p_1 \dots p_n$ for $n \geq 1$ such that $N(p_0)$, $N(p_n) = 1$, 3, or 4, and $N(p_i) = 2$ for $i = 1, 2, \dots, n-1$.

Note that a closed path $p_0 p_1 \dots p_n$ forms an unusual picture segment if $N(p_i) = 2$ for $i = 0, 1, \dots, n$ and $n \geq 4$. Two attributes—the path length and the types of terminal points—are associated with a picture segment. According to the scale of

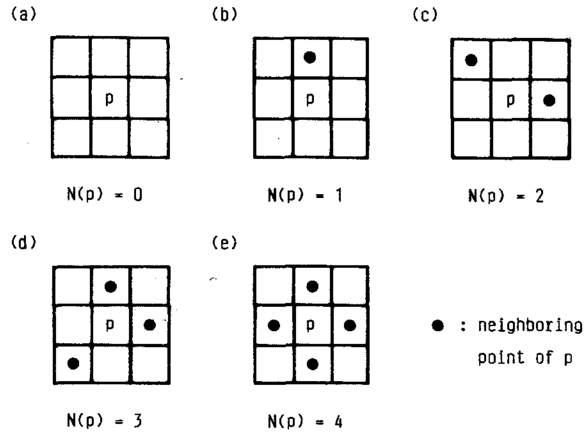


FIG. 3. Five typical types of a point p: (a) an isolated point; (b) an end point; (c) a chain point; (d) a branch point; (e) a cross point.

the path length, all the picture segments are registered as either the long or the short class with a given threshold length T_{PS}.

B. *Tracking Procedure*

In the course of tracking a picture segment, each feature point can be identified by checking the color value or computing the neighboring function of a currently visited point. All the neighboring points of a newly visited feature point are recorded for tracking the other picture segments concerned.

DEFINITION 3.6. In the light of the designations shown in Fig. 2, the *tracking direction* is defined as $D(pn_i) = i$, for a path tracked from a point p to its neighboring point n_i with $i = 0, 1, \ldots, 7$.

To avoid tracking a picture segment more than once, each currently visited point, with the exception of dividing points, is removed from the binary image. That is, the original color value is changed to 0. While, the dividing points remain to delimit picture segments, whose color values are replaced by 2, 3, and 4 for end, branch, and cross points, respectively.

ALGORITHM 1. Segment tracking.
⟨Input⟩ A visited feature point p_{F1}, together with its neighboring points $N_{F1} = \{n_{i1} \mid C(n_{i1}) \neq 0\}$.
⟨Output⟩ A picture segment S, and a newly visited feature point p_{F2}, together with its neighboring points $N_{F2} = \{n_{i2} \mid C(n_{i2}) \neq 0\}$, if any.
⟨Method⟩
1. Set $p_0 = p_{F1}$ and $i = 1$.
2. Append a point $p_i \in N_{F1}$ to p_0, and delete p_i from N_{F1}.
3. **Repeat** steps 4–7 **until** $N(p_i) \neq 2$ or $C(p_i) = 3$ or $C(p_i) = 4$.
   Begin.
4. Determine $D(p_{i-1}p_i)$.
5. Set $C(p_i) = 0$, and increase i by one.

6. Go along $D(p_{i-2}p_{i-1})$ to find such a point p_i that $C(p_i) \neq 0$.
7. Append p_i to p_{i-1}.
End.
8. Set $p_{F2} = p_i$, and return $S = p_{F1}p_1 \cdots p_{i-1}p_{F2}$.
9. **If** $C(p_{F2}) = 1$, **then do:**
Begin.
10. **Case** $N(p_{F2})$ **of**
1: set $C(p_{F2}) = 2$;
3: set $C(p_{F2}) = 3$;
4: set $C(p_{F2}) = 4$
End.
11. Return N_{F2} as well as p_{F2}.
End.
12. **End of algorithm.**

At the beginning of tracking a picture segment, a feature point is taken as the starting point and then one of its neighboring points is detected. If the detected point is also a feature point, then a picture segment whose path length equals two has been tracked. Otherwise, the initial tracking direction is determined at once. Hereafter, every following point is repeatedly selected from one of the five neighboring pixels around the last visited point, in the non-backward directions along the last tracking direction, until another feature point is visited or the starting point is visited again. If the last visited point is tracked in the direction of 4, for example, then its neighboring pixels in the directions of 2, 3, 4, 5, and 6 are taken into account on the current track.

IV. PRIMITIVE RECOGNITION

A picture segment is a basic primitive of an object. The primitive recognition process is a simple classification of picture segments based on their shapes.

A. *Piecewise Linear Approximation*

Piecewise linear approximation for describing picture segments is a compact and effective representation which retains the main structural features by virtue of succinct data sparingly.

For a given picture segment $S = p_0p_1 \ldots p_n$, the simplified piecewise linear approximation problem is defined to seek an approximated picture segment S^* which consists of a reduced path $p_0^*p_1^* \ldots p_m^*$ with an economical number of points within S. In this manner, $p_0^* = p_0$, $p_m^* = p_n$, and the jth interval of S^* constitutes the line segment $\overline{p_{j-1}^*p_j^*}$ for $1 \leq j \leq m$ by connecting the points p_{j-1}^* and p_j^* to approximate the subpath $p_a \ldots p_i \ldots p_b$ belonging to S, where $p_a = p_{j-1}^*$ and $p_b = p_j^*$. Thus the orderly points $p_1^*, p_2^*, \ldots, p_{m-1}^*$ compose the knots of the approximated picture segment S^*, and the distance from each point p_i to the associated line segment $\overline{p_{j-1}^*p_j^*}$ does not exceed a given error tolerance.

In a square grid, each picture segment can be encoded in a sequence of chain codes depending on the tracking directions. To detect the corners of a primitive, the local curvatures of a picture segment are the available pieces of structural information.

DEFINITION 4.1. Let the path $p_i p_{i+1} p_{i+2}$ be tracked in the succeeding directions of $D(p_i p_{i+1}) = V_i$ and $D(p_{i+1} p_{i+2}) = V_{i+1}$. The *trend* of $p_i p_{i+1} p_{i+2}$ is defined as

$$T(V_i, V_{i+1}) = [(V_i - V_{i+1} + 11) \bmod 8] - 3.$$

In general, the bend points that determine the corners of a primitive are located at the curvature maxima of a picture segment. By examining the local curvatures quantified in a sequence of trends $\{T_i\}$, two basic election rules for bend points are given below. Set $i = 0$ and $j = 0$ initially. The rule (a) is used for finding a candidate for a bend point after a straight path, and the rule (b) after a zigzag path.

(a) **If** $T_i = 0$, **then do**:
Begin.
If $T_{i+1} = 0$, **then** $i := i + 1$.
Else do:
Begin.
$j := j + 1$, $p_j := p_{i+2}$, and $i := i + 2$.
End.
End.

(b) **If** $T_i = r$, **then do**:
Begin.
If $T_{i+1} = -r$, **then** $i := i + 1$.
Else do:
Begin.
$j := j + 1$, $p_j := p_{i+1}$, and $i := i + 1$.
End.
End.

Note that r is only equal to ± 1 or ± 2 in the rule (b). For a short picture segment, each selected point acts as a bend point through the election process. For a long picture segment, however, the selected points are merely quasi-bend points. In this situation, further calculation with certain iterative processes must be performed on the reduced path $S^{(1)}$ that comprises the quasi-bend points from a global view point.

Consider the three consecutive points p_j, p_{j+1}, and p_{j+2} in a reduced path $S^{(q+1)}$ which is deduced from $S^{(q)}$ at the $(q + 1)$th iteration for $q \geq 1$. If these points are non-collinear, then the line segments $\overline{p_j p_{j+1}}$, $\overline{p_{j+1} p_{j+2}}$, and $\overline{p_j p_{j+2}}$ form the sides of a triangle $\Delta p_j p_{j+1} p_{j+2}$. Assuming that the line segment $\overline{p_j p_{j+2}}$ is the base of $\Delta p_j p_{j+1} p_{j+2}$, then the height of $\Delta p_j p_{j+1} p_{j+2}$ serves as the deviation for approximating the series of $\overline{p_j p_{j+1}}$ and $\overline{p_{j+1} p_{j+2}}$ to $\overline{p_j p_{j+2}}$.

DEFINITION 4.2. With elementary analytical geometry, the line segment $\overline{p_j p_{j+2}}$ can be included by the line equation

$$x(y_{j+2} - y_j) - y(x_{j+2} - x_j) + (y_j x_{j+2} - x_j y_{j+2}) = 0,$$

where x and y are the coordinates of a certain point lying on the straight line through the points p_j and p_{j+2}. The *deviation* from p_{j+1} to $\overline{p_j p_{j+2}}$ is defined as

$$d(p_j, p_{j+1}, p_{j+2}) = \left| \frac{x_{j+1}(y_{j+2} - y_j) - y_{j+1}(x_{j+2} - x_j) + (y_j x_{j+2} - x_j y_{j+2})}{\left[(x_{j+2} - x_j)^2 + (y_{j+2} - y_j)^2\right]^{1/2}} \right|.$$

Thus the point p_{j+1} within or out of the reduced path $S^{(q)}$ is declared by checking whether the deviation satisfies the given criterion or not. If $d(p_j, p_{j+1}, p_{j+2}) < \varepsilon$, where ε is a uniform error factor, then the three points p_j, p_{j+1}, and p_{j+2} are regarded as collinearity. Hence the point p_{j+1} is expelled from $S^{(q)}$, and the next collinearity check operates on the three consecutive points p_{j+2}, p_{j+3}, and p_{j+4} within $S^{(q)}$. On the other hand, the three points p_j, p_{j+1}, and p_{j+2} are non-collinear, and the point p_{j+1} is then reserved in $S^{(q)}$. In such a case, the collinearity of the three consecutive points p_{j+1}, p_{j+2}, and p_{j+3} within $S^{(q)}$ is the next to be examined.

ALGORITHM 2. Piecewise linear approximation.
⟨Input⟩ A long picture segment S in a reduced path $p_0^{(1)} p_1^{(1)} \ldots p_n^{(1)}$ for $n \geq 2$, the path length $L(S)$, and a uniform error factor ε.
⟨Output⟩ An approximated picture segment $S^* = p_0^{(q)} p_1^{(q)} \ldots p_m^{(q)}$, where $q \geq 2$, $p_0^{(q)} = p_0^{(1)}$, $p_m^{(q)} = p_n^{(1)}$, $p_j^{(q)} = p_a^{(1)}$, and $p_{j+1}^{(q)} = p_b^{(1)}$ for $0 \leq j < m$, $0 \leq a < n$, and $a < b \leq n$.
⟨Method⟩
1. Set $q = 1$, $N^{(0)} = L(S)$, and $N^{(1)} = n$.
2. **Repeat** steps 3–13 **until** $N^{(q)} = 1$ or $N^{(q)} = N^{(q-1)}$.
   Begin.
3. Increase q by one, and set $i = 1$, $j = 0$, and $d_0^{(q-1)} = 0.00$.
4. **While** $i < N^{(q-1)}$ **do** steps 5–11.
   Begin.
5. Calculate the deviation $d_i^{(q-1)}$ from $p_i^{(q-1)}$ to $\overline{p_{i-1}^{(q-1)} p_{i+1}^{(q-1)}}$, and increase j by one.
6. **If** $d_i^{(q-1)} < \varepsilon$, **then do**:
   Begin.
7. Set $p_j^{(q)} = p_{i+1}^{(q-1)}$ and $d_{i+1}^{(q-1)} = 0.00$.
8. **If** $i = N^{(q-1)} - 1$, **then** set $N^{(q)} = j$ and go to step 13.
9. Increase i by two.
   End.
10. **Else do**:
11. Set $p_j^{(q)} = p_i^{(q-1)}$, and increase i by one.
   End.
12. Set $d_i^{(q-1)} = 0.00$, $N^{(q)} = j + 1$, and $p_{N^{(q)}}^{(q)} = p_{N^{(q-1)}}^{(q-1)}$.
13. Set $p_0^{(q)} = p_0^{(q-1)}$.
   End.
14. Return $S^* = p_0^{(q)} p_1^{(q)} \ldots p_{N^{(q)}}^{(q)}$.
15. **End of algorithm.**

In this algorithm, some points are subtracted from the reduced path $S^{(q)}$ to organize a new reduced path $S^{(q+1)}$ throughout the $(q+1)$th iteration. The calculation process of judging bend points is repeatedly executed until the path length of $S^{(q)}$ is no more decreased. At this moment, the reduced path $S^{(q+1)}$ is none other than the approximated picture segment S^* of the given long picture segment S after $q+1$ iterative approximations.

B. *Primitive Classification*

The spatial noise generated from the preprocessing phase causes a few distorted or obstructed picture segments in the binary image. These irregular picture segments should be eliminated in the process of extracting essential components.

DEFINITION 4.3. Let T_{HS} be the threshold length for noise discrimination. Then a picture segment $S = p_0 p_1 \ldots p_n$ is a *hair segment*, if $L(S) \leq T_{\mathrm{HS}}$, and $N(p_0) = 1$ and/or $N(p_n) = 1$.

Some individual picture segments compose the circuit symbols, such as resistors and inductors. These objects can be easily inspected when part of a long approximated picture segment is described by a series of short vectors. For the other picture segments, each of them except the hair segments is a fundamental constituent of an essential component.

After the approximation procedure, every picture segment can be represented by a sequence of vectors encoded in chain codes pursuant to their orientations. On the basis of vector data, the primitive classification is independent of whether the translating, scaling, and rotating operations perform on the picture segments. There are six definite classes of primitives—resistors, inductors, straights, doglegs, zigzags, and curves—grouped by employing four criteria depicted as follows:

Let an approximated picture segment be symbolized by a sequence of vectors $V_a V_b \ldots V_c$, and T_{VS} be the threshold length for distinguishing the size of a vector.

Criterion 1. The existence of a long approximated picture segment which contains a series of vectors $V_1 V_2 \ldots V_k$ and $k \geq 4$ such that $V_i \in \{V_a V_b \ldots V_c\}$, and $T_{\mathrm{HS}} < L(V_i) \leq T_{\mathrm{VS}}$ for $i = 1, 2, \ldots, k$.

Criterion 2. The existence of a series of trends $T_1 T_2 \ldots T_{k-1}$ which is derived from every couple of consecutive vectors in $V_1 V_2 \ldots V_k$ such that $T_i = \pm 2$, ± 3, or ± 4 for $i = 1, 2, \ldots, k-1$, and $T_i \cdot T_{i+1} < 0$ for $i = 1, 2, \ldots, k-2$.

Criterion 3. The number of vectors $V_a V_b \ldots V_c$.

Criterion 4. The existence of a series of trends $T_1 T_2 \ldots T_m$ which is derived from every couple of ordinal odd or even vectors in $V_a V_b \ldots V_c$ such that $T_j = 0$ or ± 1 for $j \in \{1, 2, \ldots, m\}$.

Figure 4 illustrates the decision tree for assigning a picture segment to a class of primitives, where each node stands for the criterion and each leaf for the class. The picture segment constituting a resistor or an inductor can be further verified with the aid of its nearest text annotations to exclude the possibility of confusion. The primitive recognition process which includes both approximation and classification procedures is demonstrated below through an example.

EXAMPLE 1. Suppose that a long picture segment has been tracked in the direction of the arrow indicated in Fig. 5. By means of Algorithm 2 with $\varepsilon = 1$, the

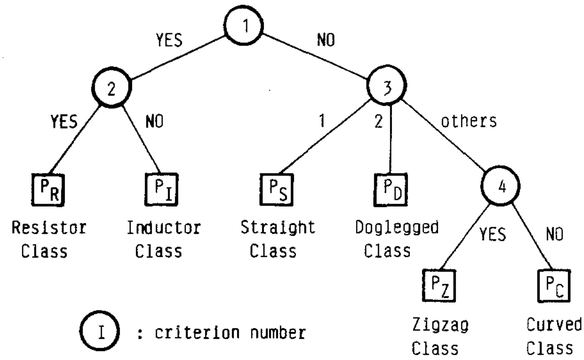


FIG. 4. Decision tree for classifying picture segments.

whole procedure for approximating the picture segment is detailed in Table 1, where the coordinates of each constitutive point are briefly denoted in order of visits. During the tracking procedure, the series of vectors $V_i^{(0)}$ and trends $T_i^{(0)}$ can be explicitly obtained. In the meantime, the quasi-bend points are selected using the election rules based on checking $T_i^{(0)}$. Henceforth, the approximating action of choosing bend points is iteratively operated according to the conditions of deviations $d_i^{(q-1)}$. After the third iteration, the remaining quasi-bend points are the same as those after the second iteration. Therefore, the approximation procedure is terminated immediately, and the picture segment is approximated to three vectors encoded as 207. The classification procedure with $T_{\mathrm{HS}} = 2$ and $T_{\mathrm{VS}} = 3$ is then follow-up, and the picture segment is registered as a curved primitive in accordance with the decision tree.

V. RELATIONAL BEST SEARCH

In a binary image, each essential component of a circuit diagram that satisfies a set of relations is embedded in an aggregation of connected graphs. A relational best search (RBS) method, which employs a depth first search technique uniting a set of specified rules, is devised to extract essential components from the binary image.

DEFINITION 5.1. A *connected graph* G within a binary image B is a maximal subset of B such that there exists a path between any two object pixels in G.

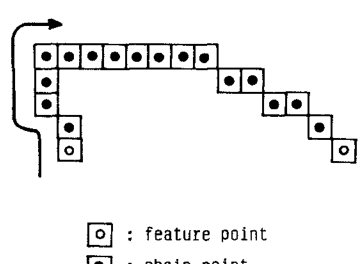


FIG. 5. Tracking locus of a picture segment.

TABLE 1
Approximation Procedure for the Picture Segment Shown in Fig. 5

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
$p_i^{(0)}$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
$V_i^{(0)}$	2	3	2	2	0	0	0	0	0	0	0	7	0	7	0	7	7	
$T_i^{(0)}$	−1	1	0	2	0	0	0	0	0	0	1	−1	1	−1	1	0	0	
$p_i^{(1)}$	0	2	4	11	15	17												
$d_i^{(1)}$	0.00	0.49	0.00	1.25	0.55													
$p_i^{(2)}$	0	4	11	17														
$d_i^{(2)}$	0.00	3.88	2.06															
$p_i^{(3)}$	0	4	11	17														

Four typical types of connected graphs distributed over a circuit diagram are exemplified in Fig. 6, where each vertex is a feature point, and each edge is a path either lying between two distinct vertices or starting and ending at the same vertex. Note that an unusual picture segment is expressed by the fourth type which has only one edge. Each picture segment within a circuit diagram is a subgraph of a connected graph.

A. *Traversal Sequence*

In the beginning of extracting essential components, the search for an object pixel except an isolated point goes forward with a 3-by-3 window in row-major order from left to right and top to bottom. Each connected graph is traversed from the first detected feature point or the first inspected point during the detection of feature points. To accelerate the extraction task, circuit symbols rather than characters are considered to be extracted above all. Since circuit symbols lie in the larger connected graphs that usually contain the main logical structure of the circuit

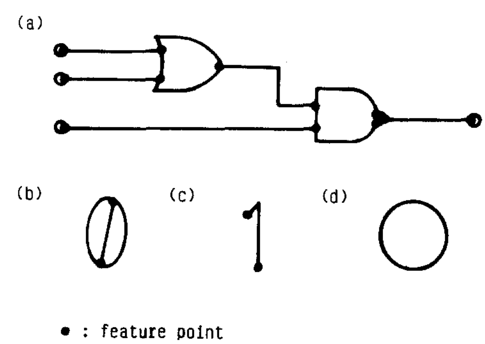


FIG. 6. Four exemplifications of the typical types of connected graphs: (a) the logical structure of a circuit diagram; (b) a numerical character "0"; (c) a numerical character "1"; (d) an alphabetical character "O".

diagram, more care should be used in detecting the large connected graphs as soon as possible.

DEFINITION 5.2. The *graph weight* denoted as $W(H)$ is measured by counting the number of object pixels within a subgraph H of a connected graph G.

For a given threshold weight T_{GS}, the scale of a connected graph can be determined by the current collected graph weight in the period of traversing the connected graph. When the connected graph is large, the process of recognizing circuit symbols is first applied for identifying each cluster of related primitives throughout the duration of the traversal; if the cluster is not a circuit symbol, the process of recognizing characters is then employed. When the connected graph is small, conversely, the process of recognizing characters is directly applied after the traversal of the connected graph. Before finding the large connected graph that contains circuit symbols, the search for other connected graphs is continued from the bottom of the last traversed connected graph in row-major order. This traversal sequence is held until the large connected graph is detected. After the large connected graph has been thoroughly traversed, the search for the remaining connected graphs starts afresh to traverse the processed binary image of the circuit diagram. Using the aforementioned traversal sequence repeatedly, all the connected graphs are completely traversed in the long run.

B. *Compound Segment Generation*

The exemplary models of circuit symbols and characters possess several qualities, such as arc and loop patterns. These qualities are available for categorizing essential components. The tracking locus of each arc or loop pattern is composed of a sequence of picture segments which may be separated by small gaps caused by a few short picture segments tracked previously.

DEFINITION 5.3. Let (x_i, y_i) and (x_j, y_j) be the coordinates of two distinct terminal points p_i and p_j of the same picture segment or two separated picture segments. The *gap* between p_i and p_j is defined as $g(p_i, p_j) = |x_i - x_j| + |y_i - y_j| + 1$.

By means of the RBS method, the search for arc and loop patterns can be accomplished step by step during the traversals of connected graphs for one time in all. Figure 7 shows the basic arc and loop patterns without gaps. At each step of the search, the picture segments meeting at or near to a feature point must be first rearranged if some of them are produced or obstructed by spatial noise. At the outset of traversing a connected graph, the first inspected point is stored in a register and regarded as a starting point to detect a feature point. If the connected graph has no feature points, it is traversed from the first inspected point anew. In the other case, the first detected feature point is substituted for the first inspected point to track all its touching or neighboring picture segments. During the tracking procedure, the order of visits decides the direction of a picture segment.

DEFINITION 5.4. A picture segment S is tracked from one terminal point denoted as **tail(S)** to the other as **head(S)**.

Note that both points tail(S) and head(S) have the same coordinates if they belong to an unusual picture segment or a picture segment S in a self-loop closed

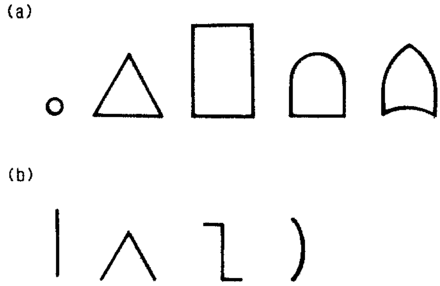


FIG. 7. Basic qualities of circuit symbols and characters: (a) loop patterns; (b) arc patterns.

with a feature point. At a branch or a cross point p_i, some picture segments meeting at p_i may be the hair segments that should be removed from the connected graphs concerned. At an end point p_j, a picture segment is near to p_j within a gap if one of its terminal points p_k separates from p_j by $g(p_j, p_k) \leq T_{\mathrm{HS}}$. Once the spatial noise meeting at p_i or near to p_j is cleared out by cutting off hair segments or making up gaps with object pixels, the neighboring value of the associated terminal point in a remaining picture segment is updated by counting the number of the remaining picture segments related to p_i or p_j. After the noise removal, the type of the terminal point in a remaining picture segment is confirmed.

Through the above-mentioned rearrangement at a search step, two touching or neighboring picture segments are selected to generate part of a compound segment with no existence of a zigzag form. In the generation of a compound segment, every selected pair of picture segments best satisfies one of the four neighboring relations that are depicted below according to priority.

Let two sequences of vectors $V_pV_q \dots V_r$ and $V_xV_y \dots V_z$, where the chain code of each vector is replaced by its residue with respect to the modulus 4, represent two distinct approximated picture segments $S_i = p_a p_b \dots p_c$ and $S_j = p_d p_e \dots p_f$ with head$(S_i) = p_c$, tail$(S_i) = p_a$, head$(S_j) = p_f$, and tail$(S_j) = p_d$, individually. The *neighboring relations* between S_i and S_j are described as follows:

(a) *Loop relation*: $g(p_c, p_f) \leq T_{\mathrm{PS}}$.

(b) *Symmetrical relation*: $\Sigma_i V_i \equiv 0 \pmod 4$ for $i = p, q, \dots, r, x, y, \dots, z$.

(c) *Smooth relation*: $T(V_p, V_x) = 0$ for $g(p_a, p_d) \leq T_{\mathrm{PS}}$, or $T(V_r, V_x) = 0$ for $g(p_c, p_d) \leq T_{\mathrm{PS}}$.

(d) *Uneven relation*: $T(V_p, V_x) \neq 0$ for $g(p_a, p_d) \leq T_{\mathrm{PS}}$, or $T(V_r, V_x) \neq 0$ for $g(p_c, p_d) \leq T_{\mathrm{PS}}$.

Note that a special compound segment consisting of only a picture segment $S = p_0 p_1 \dots p_n$ is selected first of all, if $g(p_0, p_n) \leq T_{\mathrm{PS}}$. Several concise instances shown in Fig. 8 illustrate the RBS method at a certain search step. If two pairs of picture segments both conform to the same neighboring relation best, the pair comprising a picture segment which has been part of a compound segment is selected first. Otherwise, either pair can be arbitrarily taken as the first part of a compound segment.

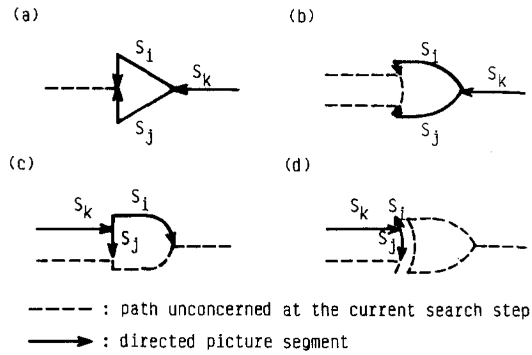


FIG. 8. A pair of picture segments S_i and S_j selected by: (a) the loop relation; (b) the symmetrical relation; (c) the smooth relation; (d) the uneven relation.

DEFINITION 5.5. Let V_i and V_j be the two neighboring vectors within an approximated picture segment or two distinct approximated picture segments. This pair of vectors is connected with either *adjacent relation*, β_1 or β_2, defined as

(a) β_1: the trend of V_i and V_j is smooth;

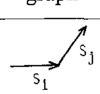
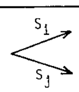
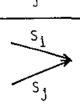
(b) β_2: the trend of V_i and V_j is uneven.

The joints in a compound segment can be characterized by the adjacent relations, which are the structural features, for determining the orientation of a loop pattern. At a search step on a feature point p_F, if head$(S_k) \neq p_F$, any picture segment S_k which fails in the election for constituting part of a compound segment is accumulated in a stack to initiate the search of another compound segment; if head(S_k) $= p_F$, it is gathered in a certain file according to its position. On the other hand, the search is continued from the point head(Q) for the compound segment that includes the current catenated pair Q in terms of picture segments S_i and S_j connected with the adjacent relation β. Table 2 shows the three catenating operations for S_i and S_j, where S_i is assumed to be in front of S_j.

The search for finding a compound segment is continuously directed by the point head(Q). This search process is interrupted when the gap between the current head(Q) and the other head(Q) reserved on another terminal of the present detected

TABLE 2

Three Catenating Operations for a Selected Pair of Picture Segments S_i and S_j

Directed graph	Condition	Action
	tail(S_j) is catenated to head(S_i)	$Q = \ldots S_i \beta S_j$ and head(Q) = head(S_j)
	tail(S_j) is catenated to tail(S_i)	$Q = S_i \beta S_j$ and head(Q) = head(S_i), head(S_j)
	head(S_j) is catenated to head(S_i)	$Q = S_i \ldots S_j \beta$ and head(Q) = head(S_i), head(S_j)

compound segment is less than the value T_{PS}, or when the type of the current head(Q) is an end point. But the process is finished when no picture segments are catenated to the last ones at all the head(Q)'s. On these suspended or terminated conditions, the preceding case is considered in preference to the succeeding one. In the first case, a compound segment is generated after the next search step is completed at the other head(Q). In the second case, the next search step is activated from the other head(Q), if any. In the final case, a compound segment has been generated. Each compound segment in the symmetrical or loop form within a certain scale is the candidate for an arc or a loop pattern, and it is ready to compose a circuit symbol or a character. A picture segment or a compound segment without the above qualities is assigned to a connection line of a signal net or a stroke of a character depending on the qualities of what arc or loop patterns it is associated with.

C. *Compound Segment Combination*

Circuit symbols may be located in the direction of 0, 2, 4, or 6, and characters in the direction of 0, 2, or 4. These essential components mostly contain arc and loop patterns which are connected with some topological relations.

DEFINITION 5.6. Let Q_i and Q_j be the two distinct compound segments, and they are connected with one of the six *topological relations* α_k for $k = 1, 2, \ldots, 6$ defined as

(a) α_1: Q_i rests on the left of Q_j;
(b) α_2: Q_i rests on the top of Q_j;
(c) α_3: Q_i rests to the left of Q_j;
(d) α_4: Q_i rests to the top of Q_j;
(e) α_5: Q_i is inside on Q_j;
(f) α_6: Q_i is inside of Q_j.

Using the RBS method, all the arc and loop patterns concerned are combined into a set of complex segments to facilitate the recognition of essential components. Throughout the traversals of connected graphs, each combination process is stimulated at the moment of starting to search for a compound segment. A small circle is an important pattern attached to the circuit symbol. In the following case, the small circle is first detected to denote the negative type of an I/O pin.

When two short picture segments satisfy the loop relation at a certain search step, the picture segment and/or the partial compound segment failing in this election is reserved in a register, and the next search step repeatedly starts from the head point of each short picture segment until all the other picture segments touching the small circle have been tracked. Hence the small circle resting on a main compound segment is found, and the succeeding search step is to initiate or resume the search of the main compound segment. At this search step, the catenating operation is performed on the individual picture segments and/or the partial compound segment that connects the small circle. In this tracking order, the main compound segment forms an arc or a loop pattern, without the small circle.

The combination process of generating a complex segment is based on combining some main compound segments, together with a few minor compound segments.

When a main compound segment is generated in the combination process, the combining operation is interrupted and then resumed to combine the other main compound segments until neither the main compound segments nor the minor ones are situated in its front or back. Because the main compound segment constitutes the major part of a circuit symbol or a character, its positional orientation guides the direction of the circuit symbol or the character.

During the traversal of a connected graph, the picture segments out of the main compound segments or small circles are the connection lines among circuit symbols or the partial constituents of characters. If the traversed connected graph includes circuit symbols, then the outside picture segments are first regarded as the connection lines for constructing equipotential signal nets. If not, they are regarded as the strokes for composing characters first. The following example utilizes a context-free grammar to illustrate the extraction process of essential components.

EXAMPLE 2. The syntactic grammar that generates the sentences to represent essential components in picture description language (PDL) is defined as

$$G_M = (V_N, V_T, R, S_C),$$

where

$V_N = \{S_C, S_L\}$ is a set of nonterminal symbols;

$V_T = \{(\ ,\ )\} \cup \{\alpha\} \cup \{\beta\} \cup \{\gamma\}$ is a set of terminal symbols;

$R = \{S_C \to (S_C \alpha S_C),\ S_C \to S_L,\ S_L \to (S_L),\ S_L \to S_L \beta S_L,\ S_L \to S_L \beta,\ S_L \to \gamma\}$ is a set of production rules;

S_C is a starting symbol.

Note that $\alpha \in \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6\}$, $\beta \in \{\beta_1, \beta_2\}$, and γ may be any primitive including the null primitive λ; that is, $\gamma \in \{P_R, P_I, P_S, P_D, P_Z, P_C, \lambda\}$.

Figure 9(a) shows a partial schematic drawing of a circuit diagram, which consists of three connected graphs. According to the principles of the RBS method, a substructure of the control diagram given in Fig. 9(b) enforces the production rules to be applied as many times as possible. In such a case, the traversals of text annotations are preceded by that of the logical structure of the circuit diagram. After these traversals, the three connected graphs are grouped into the three categories of essential components expressed by PDL and shown in Fig. 9(c). The PDL expression for the cluster of a circuit symbol or a character is an infix-type expression.

VI. EXPERIMENTS

An automatic understanding system has been developed for comprehending electronic circuit diagrams by a computer. In the present system, the whole processing software is implemented in FORTRAN under the VMS3.4 operating system of a super minicomputer VAX-11/780. The program size is about 12,000 lines of codes.

Seven machine-printed and three hand-sketched circuit diagrams from different areas of electronic engineering have been used for testing the component extractor. These circuit diagrams comprise 23 varied types of circuit symbols, where the total

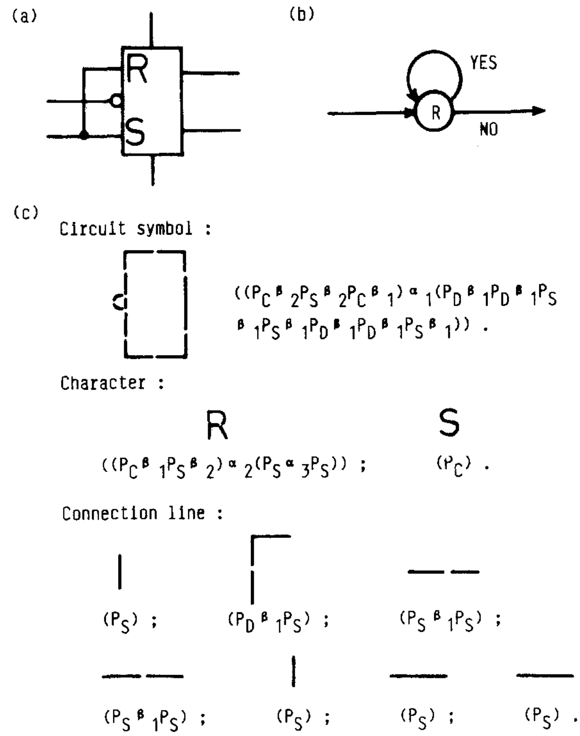


FIG. 9. (a) An R–S flip-flop with seven I/O pins; (b) a simple control diagram for employing production rules; (c) subgraphs with PDL expressions extracted from the circuit diagram (a).

number of circuit symbols is 418 and that of text annotations consisting of 1958 characters is 796. In these experiments, only a few essential components are unrecognized owing to the failures in separating the connections between characters and other essential components. The recognition rate of the extracted essential components is more than 98% on an average. For the unidentified essential components, they can be recognized in a hand-digitizing approach via an interactive graphic editor supplied by the present system.

Shown in Fig. 10 is an experimental example of a hand-sketched circuit diagram. The circuit diagram drawn on a sheet of paper of the size 19×23 cm^2 is entered into the computer through a drum scanner with the resolution 7 pixels/mm. In this experiment, the error factor used in the component extractor is given $\varepsilon = 5$, and the threshold values are $T_{PS} = 25$, $T_{HS} = 5$, $T_{VS} = 7$, and $T_{GS} = 250$, respectively. The component extractor runs the binary image of Fig. 10 on 180 Kbyte, and the execution time for this example takes less than 20 min., including preprocessing operations. Figure 11 demonstrates the accurate understanding result of Fig. 10 in a fair copy which is reproduced by the present system.

VII. DISCUSSIONS AND CONCLUSIONS

This paper has presented a topology-based component extractor for the understanding of electronic circuit diagrams. Each essential component comprises a set of picture segments located among feature points and appropriately detected by a

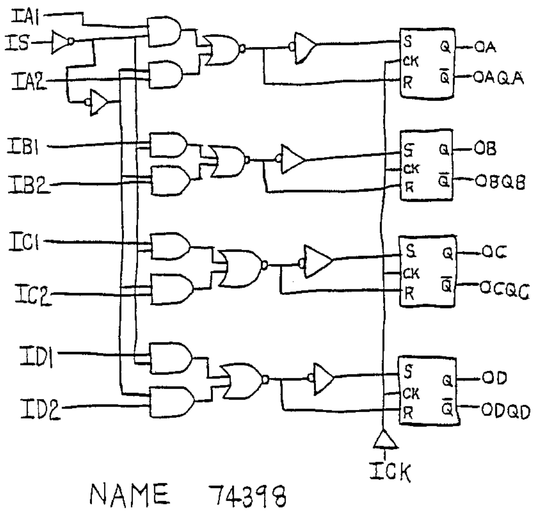


FIG. 10. Hand-sketched circuit diagram of a quadruple 2-input multiplexers with storage.

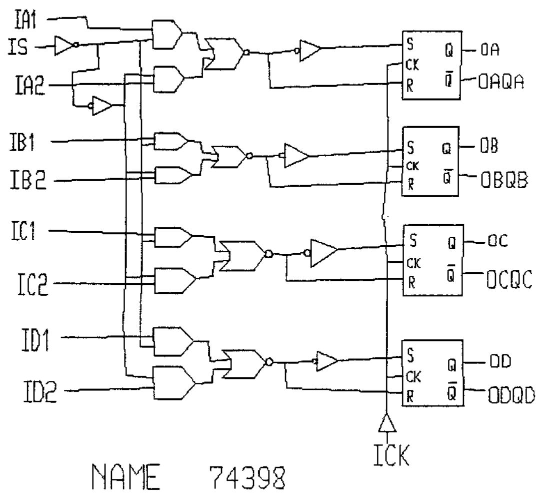


FIG. 11. Fair copy of the circuit diagram shown in Fig. 10 after recognition.

segment tracking algorithm. A simplified piecewise linear approximation problem is set up for describing a picture segment by the economical number of approximated segments, and it is solved by a heuristic algorithm to speed up the recognition of primitives. For a given error factor, the approximation result derived from our algorithm is preferable to that from the optimum algorithm [14], and the execution time of our algorithm is less than that of the fast algorithm [15]. A detailed comparison of the approximation results obtained from these two prominent algorithms and ours has been proposed in the literature [16].

The most important and powerful concept of the component extractor is based on the topological analysis of the context of picture segments during the traversal of a circuit diagram in one pass. A relational best search method realizes the concept by use of the depth first search technique uniting a set of specified rules in accordance with a prior knowledge about the diagram models. Once the traversal of the circuit diagram is accomplished, all the essential components are extracted and instantly grouped into three categories in terms of circuit symbols, characters, and connection lines for interpreting circuit functions in the subsequent processing phases.

Subject to the resolution of an input image is the error factor used for approximating picture segments as well as the threshold values for classifying the scales of picture segments, vectors, connected graphs, and so forth. However, these measures can be learned from a pilot study on the data analysis by means of the statistical decision theory [17]. Under a fixed image resolution, the learned measures are suitable for treating the various sizes of essential components independent of the scale and the complexity of a particular circuit diagram.

Throughout the extracting phase, each classified essential component is registered by the identification and position data. Consequently, the original image data of a circuit diagram are compressed into a quantity which will be easily handled in the later processing phases for most contemporary computer systems. On the average, the data compression ratio is about 0.05 after the extracting phase.

Of the present system, the execution time in the preprocessing phase dominates most of the computational term. To improve the processing efficiency, the preprocessing operations implemented by the hardware technique are a realizable way [18]. As usual, the spatial noise produced in the preprocessing phase may interfere with the objects lying in the binary image. In our manner, nevertheless, the noise is deleted from the image phase by phase.

The principles of the topology-based component extractor are adequate enough to understand many other types of schematic drawings, for example, IC layout diagrams, process flow charts, geographic maps, and piping and instrument diagrams, by only replacing the special models of graphic symbols and the special connection rules among the graphic symbols.

APPENDIX: LIST OF SYMBOLS

$C(\cdot)$, color value of a pixel.
$L(\cdot)$, length of a path.
$N(\cdot)$, neighboring value of a point.
$D(\cdot)$, tracking direction of a path from a point to its neighbor.
$W(\cdot)$, weight of a subgraph of a connected graph.
tail$(\cdot)$, tail point of a picture segment or a compound segment.
head$(\cdot)$, head point of a picture segment or a compound segment.
$T(V_i, V_j)$, trend between two vectors V_i and V_j.
$g(p_i, p_j)$, gap between two points p_i and p_j.
$d(p_i, p_j, p_k)$, deviation from a point p_j to a line segment $\overline{p_i p_k}$.
T_{PS}, threshold length for classifying the scale of a picture segment.
T_{VS}, threshold length for classifying the scale of a vector.
T_{HS}, threshold length for noise discrimination.
T_{GS}, threshold weight for classifying the scale of a connected graph.
ε, error factor for approximating picture segments.
β, adjacent relation between two vectors.
α, topological relation between two compound segments.

REFERENCES

1. R. Rutman, Non-gridded graphic input, in *Proceedings, 13th Design Automation Conf., 1976*, pp. 392–398.
2. H. M. Bayegan, CASS: Computer aided schematic system, in *Proceedings, 14th Design Automation Conf., 1977*, pp. 396–404.
3. H. M. Bayegan, An integrated system for interactive editing of schematics logic simulation and PCB layout design, in *Proceedings, 15th Design Automation Conf., 1978*, pp. 1–8.
4. T. M. McWilliams and L. C. Widdoes, Jr., SCALD: Structured computer-aided logic design, in *Proceedings, 15th Design Automation Conf., 1978*, pp. 271–277.
5. S. Kakumoto *et al.*, Logic diagram recognition by divide and synthesize method, in *Artificial Intelligence and Pattern Recognition in Computer Aided Design* (J. C. Latombe, Ed.), pp. 457–470, North-Holland, Amsterdam, 1978.
6. M. Ishii *et al.*, An experimental input system of hand-drawn logic circuit diagram for LSI CAD, in *Proceedings, 16th Design Automation Conf., 1979*, pp. 114–120.
7. B. Zavidovique and G. Stamon, An automated process for electronics scheme analysis, in *Proceedings, 5th Int. Conf. Pattern Recognition, 1980*, pp. 248–250.
8. M. Ishii *et al.*, Automatic input and interactive editing systems of logic circuit diagrams, in *Proceedings, 18th Design Automation Conf., 1981*, pp. 639–645.
9. T. Sato and A. Tojo, Recognition and understanding of hand-drawn diagrams, in *Proceedings, 6th Int. Conf. Pattern Recognition, 1982*, pp. 674–677.
10. J. T. Tou and J. M. Cheng, AUTORED: An automated electronic diagram reading machine, in *Proceedings, IEEE Int. Conf. Computer-Aided Design, 1983*, pp. 19–21.
11. Y. Fukada, A primary algorithm for the understanding of logic circuit diagrams, *Pattern Recognit.* **17**, 1984, 125–134.
12. H. Bley, Segmentation and preprocessing of electrical schematics using picture graphs, *Comput. Vision Graphics Image Process.* **28**, 1984, 271–288.
13. X. Lin *et al.*, Efficient diagram understanding with characteristic pattern detection, *Comput. Vision Graphics Image Process.* **30**, 1985, 84–106.
14. J. G. Dunham, Optimum uniform piecewise linear approximation of planar curves, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-8**, 1986, 67–75.
15. J. Roberge, A data reduction algorithm for planar curves, *Comput. Vision Graphics Image Process.* **29**, 1985, 168–195.
16. C. S. Fahn *et al.*, An adaptive reduction procedure for the piecewise linear approximation of digitized curves, *IEEE Trans. Pattern Anal. Mach. Intell.*, in press.
17. A. M. Law and W. D. Kelton, *Simulation Modeling and Analysis*, McGraw–Hill, New York, 1982.
18. K. S. Fu, *VLSI for Pattern Recognition and Image Processing*, Springer-Verlag, Berlin/ Heidelberg/ New York, 1984.