Recognition of Electrical Symbols in Document Images Using Morphology and Geometric Analysis

Paramita De
Computer Science & Engineering
Haldia Institute of Technology
India
Email: paramitade13@gmail.com

Sekhar Mandal
Computer Science & Technology
BESU Shibpur
India
Email: sekhar@cs.becs.ac.in

Partha Bhowmick
Computer Science & Engineering
IIT Kharagpur
India
Email: bhowmick@gmail.com

***Abstract*—A robust and efficient algorithm for recognition of electrical symbols in digitized documents is proposed. The algorithm is based on morphological operations and geometric analysis to recognize different classes of symbols. Its novelty lies in building and usage of three spaces, namely H-, V-, and C-spaces, which respectively contain the horizontal line segments, the vertical line segments, and the circuit symbols present in the concerned drawing. These three spaces are built by morphological operations, which, in turn, are searched and scanned in a scientific way during the geometric analysis in order to obtain the recognized symbols by verifying the structural combination of their constituent primitives. Exhaustive experimentation has been performed to test the performance of the algorithm. Some of the results are presented in this paper to demonstrate its robustness, efficiency, and versatility.**

***Keywords*-Electrical symbols, morphological operation, geometric analysis, symbol recognition.**

I. Introduction

A significant progress can be noticed—starting from 1980's and recently with increasing demands—on the processing and understanding of digitized engineering drawings [7]. Some of the approaches are off-line [1,2,6,15] and some are online [9, 11, 15]. Out of all these existing approaches, however, quite a few have focused on recognition of symbols in an electrical drawing present in the input image of a digitized document after segmenting out the text from the graphics. In particular, to the best of our knowledge, there is no published work till date, which has proposed an integrated approach to recognize different classes of electrical symbols from digitized documents. For example, the earlier approaches in [4,8,12] consider only logic circuit diagrams (made of OR, AND, and inverter) as input; other electrical symbols (Sec. II) are not taken into consideration in the input drawings. Features and restrictions of later works are briefed below.

In [2], a symbol recognition system is proposed based on *neural network*. Moments are taken as features and a hierarchical *multi-layer perceptron* with *back propagation learning* algorithm is implemented. It is assumed that all the symbols must have same number of features invariant to their orientation, size, and location in the entire drawing. The algorithm in [1] is designed only for a DC and not for an AC circuit, and uses *moment-invariant descriptors*. The online scheme in [15] uses the *spatial arrangement* identified from the strokes while drawing. Based on a dictionary of features, a component is compared with known components to assign *confidence factors*. In [6], *probabilistic graph matching* is used with where the skeletons of symbols are represented by graphs to match the class models. A graph-based system for recognizing a large class of engineering drawings is also presented in [16]. It has, however, adopted a generalization of input, e.g., flowcharts and chemical plant diagrams, and not focused in the exhaustiveness of all classes of electrical symbols.

Our work, on the contrary, is based on certain *morphological operations*, which, in turn, help in implementing a novel idea of three spaces containing the horizontal line segments (*H-space*), vertical line segments (*V-space*), and (possibly broken) circuit symbols (*C-space*). These three spaces are built without thinning, and used in an efficient way for searching and scanning during the *geometric analysis* so as to derive the necessary relation among the constituent *primitives* of the symbols to be recognized using their structural features, as explained next.

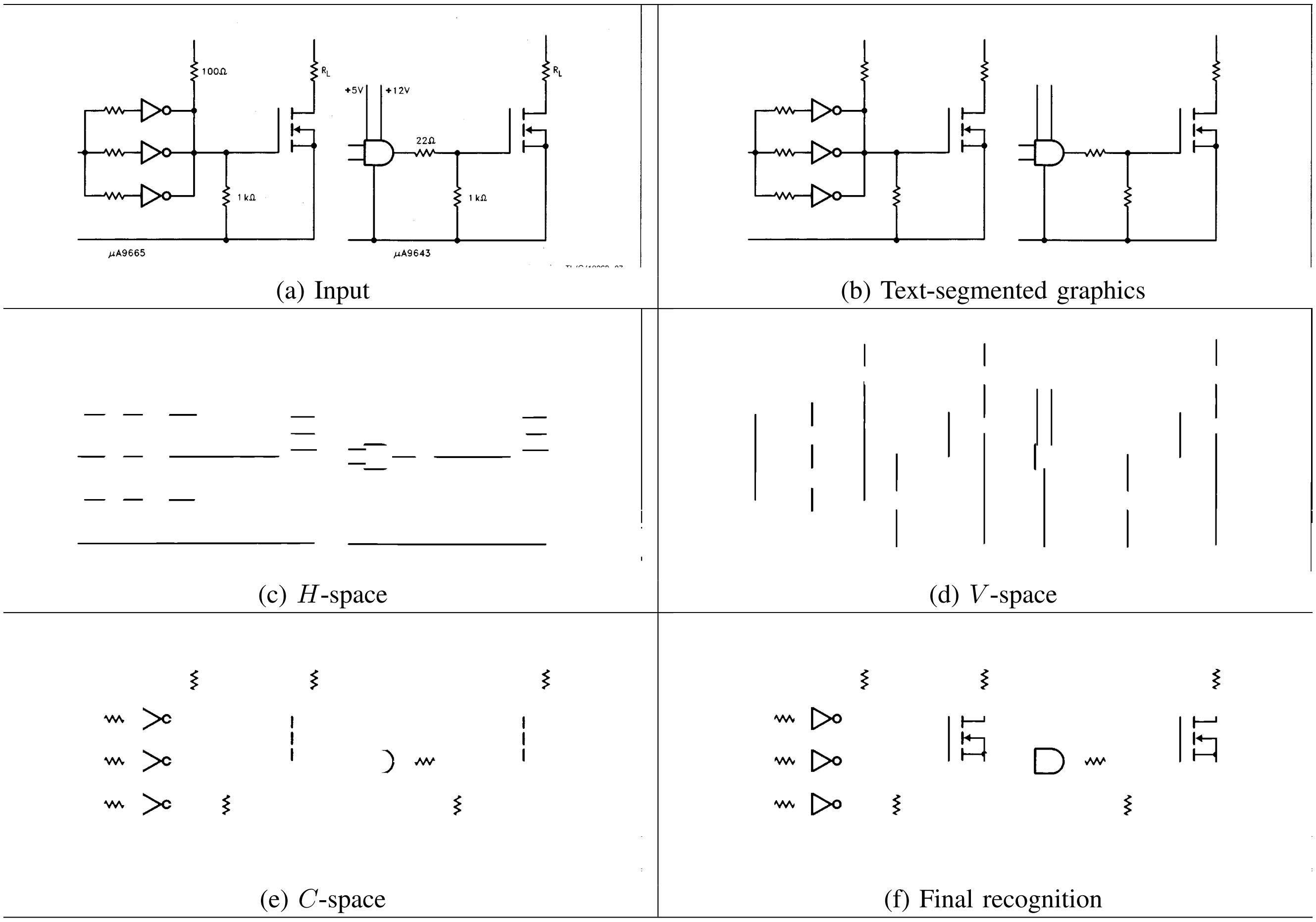
II. Symbols and their Properties

The acronyms and the typical digital images of the symbols being given in Appendix, we briefly explain below their structural properties w.r.t. their first orientation in Appendix. Other orientations (2, 4, or 8 in number) are by rotation or flip of the first orientation, and hence characterized by equivalent properties. These properties are used for their detection, as explained in Sec. III.

- **RES** A *resistor* is symbolized by three or more zigzag-cum-periodic strokes of lines.[1]
- **IND** An *inductor* closely resembles RES apropos its periodic strokes, which are, distinguishingly, constituted by spiral lines.
- **TRF** A *transformer* consists of two oppositely faced INDs with two or more equi-length line segments between them.

[1] Here and henceforth in this paper, "line" means "straight line".

978-1-61284-861-7/11/$26.00



(a) Input

(b) Text-segmented graphics

(c) H-space

(d) V-space

(e) C-space

(f) Final recognition

Figure 1. Step-by-step demonstration of our algorithm on a portion of a typical document page containing an electrical drawing.

- **CAP** A *capacitor* has two plates symbolized by a pair of parallel line segments, or a combination of a line segment and a concave segment of equal height.[2]
- **BAT** A *battery* has a constitution similar to CAP with two parallel line segments, excepting that one segment is longer than the other.
- **MOS** An NMOS and a PMOS have a common structural characteristic: Each has two vertical equi-length parallel line segments, one of whose both endpoints are incident on two line segments (technically called *source* and *drain*). The other line segment (*gate*) has either a line incident at its midpoint (NMOS) or a circle touching its midpoint (PMOS).
- **FET** A *field effect transistor* has a vertical line segment (*gate*) and three small-and-collinear line segments (*source*, *substrate*, *drain*) parallel to gate. The distance between the farthest endpoints of source and drain equals the gate length.
- **TRN** A *transistor* essentially consists of three line segments (*base*, *emitter*, and *collector*), which may or may not be lying in an enclosing circle. The emitter and the base are so incident on the base that the acute angles formed by them with the base are equal.
- **INV** An *inverter* consists of a triangle with one vertical edge, and two other equi-length edges whose common vertex touches a small circle.
- **OPA** An *operational amplifier* is similar to INV in the sense that it contains a triangle, shaped like that of an INV. But it does not contain any circular shape, which marks its difference with INV.
- **DIO** A *diode* also contains a triangular shape like OPA with its vertical edge symbolizing the *anode*. In addition, it contains a vertical line segment (*cathode*) that touches the vertex opposite to anode and has length nearing that of anode.
- **AND** Characterized by its vertical edge whose two endpoints are incident on two equi-length horizontal line segments. The other two endpoints of the horizontal line segments are connected by a convex (possibly circular) curve segment.
- **NAND** Has one component just like AND gate and, in

[2] Equality of length, width, or angle is verified in our implementation by considering a small tolerance; e.g., $l \approx l'$ if $|l - l'| \leq \frac{1}{8}\max\{l, l'\}$. This takes care of pruning and digitization error.

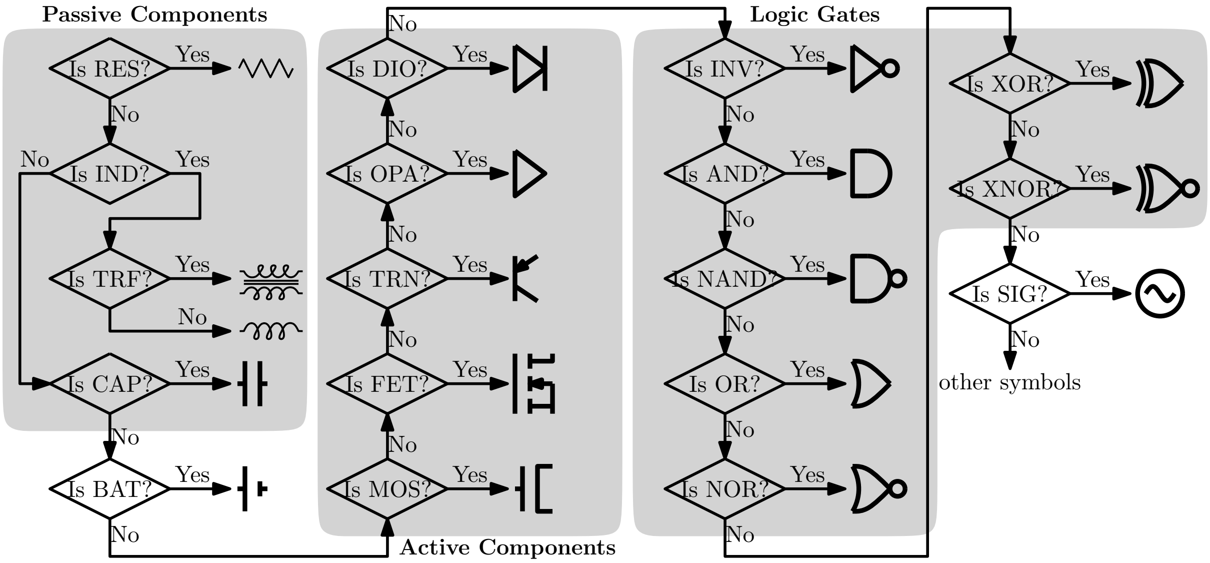


Figure 2. Proposed scheme on step-by-step detection of different symbols.

addition, a small circular shape touching the convex segment of the AND-like component.

- **OR** Has a concave edge whose two endpoints lie on a hypothetical vertical line and are incident on two equi-length horizontal line segments. The other two endpoints of the horizontal line segments are connected by a convex curve segment.
- **NOR** One OR-like component and a small circular shape touching its convex segment.
- **XOR** One OR-like component and a concave curve segment lying parallel to and of equal length with its concave segment.
- **XNOR** One NOR-like component and a curve segment lying parallel to and of equal length with its concave segment.
- **SIG** Symbolized by a sinusoidal curve segment lying inside an enclosing circle.

III. Proposed Algorithm

Figure 1 demonstrates the basic steps of our algorithm. Figure 2 shows the schematic flow of recognizing various symbols in stages, as explained below.

A. Preprocessing

The input gray-scale image is first binarized and skew-corrected using the algorithms reported in [3, 13]. The input consists of electrical circuits along with text regions. Hence, component analysis is done first, and then the median height $\widetilde{h}$ and the median width $\widetilde{w}$ are computed for each component.[3] As an electrical circuit usually forms a big component, the text portions (small components) are easily removed from the document image by comparing their heights and widths with $\widetilde{h}$ and $\widetilde{w}$. Horizontal (vertical) line segments are separated using the morphological opening operation with a structuring element of size $2\widetilde{h}\times 2$ $(2\times 2\widetilde{w})$ [5]. We have used three spaces, named as H-, V-, and C-spaces, which store the respective horizontal line segments, vertical line segments, and the circuit elements. The length of each line segment stored in the corresponding H- or V-space, is at least $2\widetilde{h}$ or $2\widetilde{w}$.

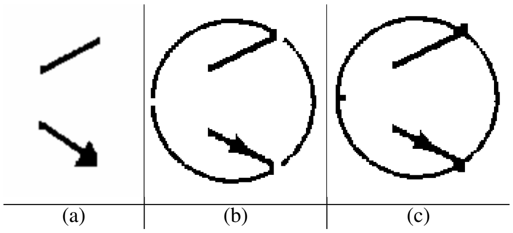


Figure 3. Inclined line segments of a transistor (TRN) lying in C-space. (a) Two primitives without enclosing circle. (b) Three components constituted by two primitives and split-off arcs of the enclosing circle. (c) A single component after joining the three components in (b).

Due to extraction of horizontal and vertical line segments while constructing the H- and the V-spaces, some of the components in C-space get disconnected. To restore the connectedness, we apply a joining procedure in C-space using the information from the H- and the V-spaces. For joining, we search in C-space from both ends of a component c within a space of $(w+2)\times(w+2)$, for its broken component c' (if any) in C-space, w being the maximum width of horizontal/vertical lines in H-/V-space. If found, then a search is performed in V-(H-)space to verify the existence of a vertical (horizontal) piece that fills the gap between c and c'; in such case, c and c' are joined. For example, in

[3] These median values signify the size of text, which are usually present—as found by us in our exhaustive experimentation on various datasets—in an electrical drawing as annotations or specifications.

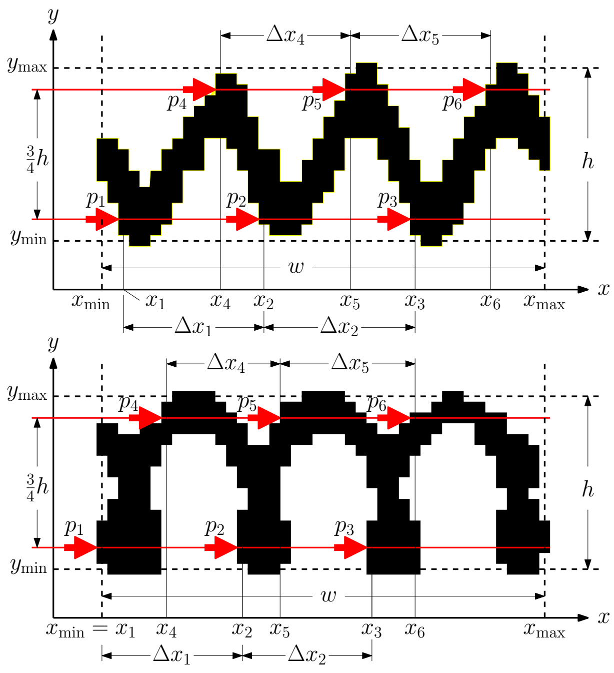

y
$y_{\max}$
$y_{\min}$
$\frac{3}{4}h$
h
w
x
p_1 p_2 p_3 p_4 p_5 p_6
$x_{\min}$ x_1 x_4 x_2 x_5 x_3 x_6 $x_{\max}$
Δx_1 Δx_2 Δx_4 Δx_5
$x_{\min} = x_1$

Figure 4. Horizontal scanning of a circuit element, which results in distinctive recognition of a resistor (top) and an inductor (bottom).

Fig. 3(a), the two primitives (for TRN) remain unchanged after joining, whereas the three disconnected components in Fig. 3(b) become a single component after joining, as shown in Fig. 3(c).

In preprocessing, we detect and store the relevant information of the primitives in C-space, which include inclined line segment, circle, and concave (convex) segments, as follows.

Inclined line segment: We do vertical (horizontal) scans from left to right (from top to bottom) and count the number of black pixels for each scan. If these counts are nearly equal, then a hypothetical line segment joining the two centroids of the black points corresponding to the two extreme vertical scan lines, is considered. If the points lying on the hypothetical segment are black, then the segment is identified as a line segment, and its length and angle of orientation are stored in C-space. The scans are performed before joining, since two inclined segments may get joined in the joining phase (e.g., OPA, DIO, etc.), resulting to difficulty in recognizing them as inclined segments.

Circles: Detected after joining phase. We consider the bounding box (BB) of a connected component in C-space and verify whether the BB is squarish. If so, we consider four scan lines whose orientations are 0^0 (horizontal), 45^0, 90^0 (vertical), and 135^0—each passing through the center of BB. We scan BB from its exterior to verify whether the distance (measuring the inner diameter of the circle) from the point of black-to-white transition to the point of white-to-black transition along each scan line nearly equals that of each other.

Concave segments: Also detected after joining phase using the BB. The orientation is decided by the height-width ratio of BB. For vertical orientation, horizontal scan lines from top to bottom are considered. Along each scan line, we measure the distance of the white-to-black transition point from the left boundary of BB. If in the ordered set of distances corresponding to the scan lines, the distances first increase and then decrease, then the component is concave from left; otherwise, if they first decrease and then increase, then the component is concave from right. A concave segment may have four types of concavities (left, right, top, bottom), which are stored in C-space.

B. Passive Components

Within the C-space, the height and the width of each component are computed. Figure 4 demonstrates the process. Let $x_{\min}$ and $x_{\max}$ be the respective x-coordinates of the leftmost and the rightmost points of a component; and let $y_{\min}$ and $y_{\max}$ be the respective y-coordinates of its bottommost and topmost points. Then its respective width and height are given by $w = x_{\max} - x_{\min}$ and $h = y_{\max} - y_{\min}$. If the ratio of width and height (height and width) is greater than 2, then two horizontal (vertical) scans are done from $x = x_{\min}$ ($y = y_{\min}$) to $x = x_{\max}$ ($y = y_{\max}$). One horizontal scan is done with $y = y_{\min} + \frac{1}{8}h$ (*bottom scan*) and another with $y = y_{\max} - \frac{1}{8}h$ (*top scan*). The number of white to black transitions (wb) is recorded for each scan. Figure 4(top) shows that for the bottom scan, wbs are at p_1, p_2, and p_3, and for the top scan, they are at p_4, p_5, and p_6.

Let the respective number of wbs for the bottom and the top scans be $t_{\min}$ and $t_{\max}$. Let x_i be the x-coordinate of p_i for $i = 1, 2, \ldots, 6$ (Fig. 4). Let $\Delta x_i = x_{i+1} - x_i$ for $i = 1, 2, 4, 5$. Let $\Delta x_{\min} = \min\{\Delta x_i : i = 1, 2, 4, 5\}$ and $\Delta x_{\max} = \max\{\Delta x_i : i = 1, 2, 4, 5\}$. If $t_{\min} = t_{\max} \geq 3$ and $\Delta x_{\max} - \Delta x_{\min} \leq \frac{1}{8}\Delta x_{\min}$, then the symbol is either RES or IND. To distinguish one from the other, the number of black pixels (n_b) is compared with that of white pixels (n_w) for the top scan. If $n_b < n_w$, then the symbol is recognized as RES; otherwise, IND (Fig. 4). For each IND, we have searched from the midpoint of (and in an orthogonal direction to) its top scan line up to a length equal to half the length of IND. The search is carried on in C-space. If any other matching IND is found and there exist two or more parallel line segments in V-space or H-space (depending on the orientation of INDs) in between them whose lengths are nearly equal to the length of INDs, then the IND pair including the parallel line segments sandwiched between them are considered as a transformer (TRF).

For a CAP formed by two vertical (horizontal) parallel lines, then we have to search in the V-(H-)space. For these two segments, if (c1) their lengths lie in $[2\widetilde{h}, 4\widetilde{h}]$ and (c2) there exists a line segment in H-space lying left (right) of and touching the midpoint of left (right) segment, then the the two components under consideration form a CAP.

If one of the two plates is a curve segment, then it lies in C-space and we work with its bounding box to verify the above conditions.

A DC power source (BAT) is denoted by two parallel line segments. As the length of one line segment is less than $2\widetilde{h}$, it will be in C-space. The other steps of BAT identification are similar to those of CAP identification.

C. Active Components

The main structure of an NMOS being similar to CAP (Sec. II), we verify condition (c1) and an appropriate modification of condition (c2) of CAP. For PMOS, in condition (c2), we search and verify the existence of a small circle (in C-space: Sec. III-A) touching the midpoint of one of the line segments constituting the PMOS.

For an FET, its *source*, *substrate*, and *drain* (Sec. II) are three small segments lying in C-space and the fourth segment in $V(H)$-space. We verify whether the combined length of the small segments (distance between the top endpoint of the upper segment and the bottom endpoint of the lower segment) equals the length of the fourth segment. We also verify the nature of incidence of the other line segments (symbolizing the wirings) by searching in V- and H-spaces to ensure the configuration of FET (Sec. II).

A TRN consists of three line segments, one of which is vertical (horizontal), lying in V-(H-)space, symbolizing the *base*, and with the other two segments lying in C-space, inclined and incident on it (Sec. II). If TRN has no enclosing circle, then we detect one inclined segment in C-space, its matching vertical segment in V-space, and matching inclined segment in C-space—in that order. If TRN has an enclosing circle, then a search is performed in V-(H-)space within the circle to find a vertical (horizontal) line segment, which, if touches the circular component at two points (Fig. 3), ensures that the element is TRN.

An OPA is detected by its triangular shape made of one vertical (horizontal) line segment in V-(H-)space and two equi-length inclined line segments in C-space. To recognize DIO, an additional search is done in V-space for the vertical line segment touching the junction point of the two inclined segments.

D. Logic Gates

An INV is recognized by its triangle and a small adjoining circle. The vertical edge of the triangle is searched in V-space from the wider side of the component, and its other two edges with the circle as a single component is checked in C-space (Fig. 5). To detect the meeting point of the triangle and the circle, we divide the component by four vertical lines into five equal parts, and scan the region between the dividing lines 3 and 4. On finding a region having exactly one transition point from black to white, we separate out the circle from the triangle at the rightmost point where number of black points (n_b) lying on the scan line is minimal. Circularity test is done as explained in Sec. III-A.

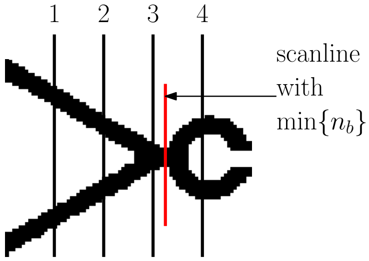


Figure 5. Detecting an inverter in C-space.

An AND gate is recognized starting from its convex segment from C-space and the matching line segments from H- and V-spaces. For NAND gate, the component in C-space consists of a convex piece and a small adjoining circle, whose "combination" is also detected as a convex piece in C-space. The matching line segments are searched as in AND gate. Separation and verification of the circle is done as in INV. An OR gate is recognized by its one concave (C-space), one convex (C-space), and two matching lines (H-space). The NOR gate is distinguished from the OR by its adjoining circle, which is verified as explained above. XOR and XNOR gates are extensions of OR and NOR by a concave piece, which is searched and verified in C-space. A SIG is identified by its enclosing circle and a disconnected sinusoidal segment lying inside the circle in C-space. The sinusoidal segment is recognized by breaking it into two equal-length pieces, and then verifying whether the two pieces have opposite concavities (top and bottom).

IV. Results and Conclusion

We have implemented our algorithm in C on an Intel(R) Core(TM)2 Duo CPU E4500 2.20 GHz machine (OS: Linux Mandriva Release 2008). We have tested it on various digitized documents scanned from different books and periodicals [10, 14]. As explained in Sec. III, the morphological operations and geometric analysis in H-, V-, and C-spaces are successfully performed on these document sets as evident in our experimental results, a few of which are presented in Figs. 1 and 6.

The input image (1121 × 1870 pixels) in Fig. 6 is a typical instance of a document page having figure caption along with bold and normal text. By our segmentation algorithm, we successfully obtain the engineering drawing, the resultant H-, V-, and C-spaces, from which the symbols are subsequently analyzed and reported; total CPU time is 2.174 secs. Detection of some electrical symbols like earthing and multiple series-connected batteries are not included in our implementation, which can, however, be done with appropriate geometric analysis.

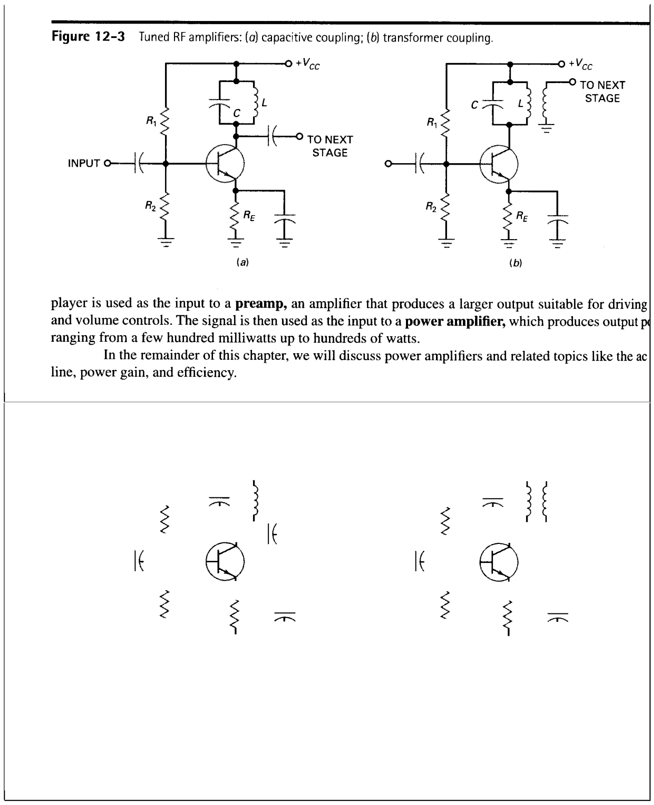


Figure 12-3 Tuned RF amplifiers: (*a*) capacitive coupling; (*b*) transformer coupling.

player is used as the input to a **preamp,** an amplifier that produces a larger output suitable for driving
and volume controls. The signal is then used as the input to a **power amplifier,** which produces output p
ranging from a few hundred milliwatts up to hundreds of watts.

In the remainder of this chapter, we will discuss power amplifiers and related topics like the ac
line, power gain, and efficiency.

Figure 6. Results by our algorithm. Top: Input. Bottom: Output.

References

[1] M. Boraie and A. Balghonaim. Optical recognition of electrical circuit drawings. In *IEEE Pacific Rim Conference Networking*, pp. 843–846, 1997.

[2] T. Cheng, J. Khan, H. Liu, and D. Yun. A symbol recognition system. In *Proc. ICDAR*, pp. 918–921, Oct. 1993.

[3] A. K. Das and B. Chanda. A fast algorithm for skew detection of document images using morphology. *IJDAR*, 4:109–114, 2001.

[4] Y. Fukada. A primary algorithm for the understanding of logic circuit diagrams. *Pattern Recognition*, 17:125–134, 1984.

[5] R. C. Gonzalez and R. Wood. *Digital Image Processing*. Addision-Wesley, 1992.

[6] F. C. Groen, A. C. Sanderson, and J. F. Schlag. Symbol recognition in electrical diagrams using probabilistic graph matching. *PRL*, 3(5):343–350, 1985.

[7] S. Joseph. Processing of engineering line drawings for automatic input to CAD. *Pattern Recognition*, 22(1):1–11, 1989.

[8] S. Kim, J. Suh, and J. Kim. Recognition of logic diagrams by identifying loops and rectilinear polylines. In *Proc. ICDAR*, pp. 349–352, 1993.

[9] H. Kojima and T. Toida. Online hand-drawn line-figure recognition and its application. In *Proc. ICPR*, pp. 1138–1142, 1988.

[10] M. Mano. *Digital Logic and Computer Design*. Pearson Prentice Hall, 3rd Ed., 2006.

[11] H. Murase and T. Wakahara. Online hand-sketched figure recognition. *Pattern Recognition*, 19(2):147–160, 1986.

[12] A. Okazaki, S. Tsunekawa, T. Kondo, K. Mori, and E. Kawamoto. An automatic circuit diagram reader with loop-structure-based symbol recognition. *IEEE Trans. PAMI*, 10:331–341, 1988.

[13] N. Otsu. A threshold selection method from gray-level histogram. *IEEE Trans. SMC*, pp. 62–66, 1979.

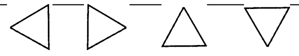
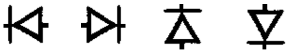
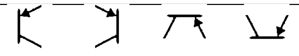
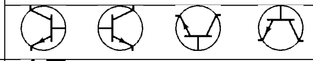
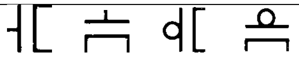
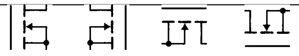
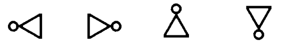
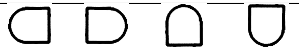
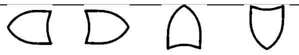
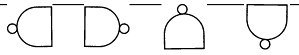
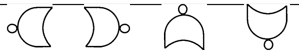
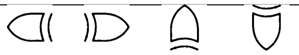
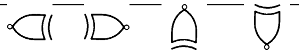
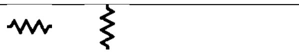
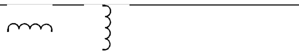
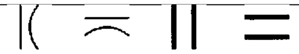
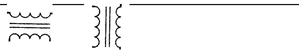
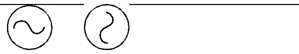
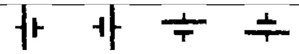
[14] D. A. Pucknell and K. Eshraghian. *Basic VLSI Design*. PHI, 3rd Ed., 2000.

[15] J.-P. Valois, M. Côté, and M. Cheriet. Online recognition of sketched electrical diagrams. *ICDAR*, pp. 460–464, 2001.

[16] Y. Yu, A. Samal, and S. Seth. A system for recognizing a large class of engineering drawings. *IEEE Trans. PAMI*, 19(8):868–890, 1997.

Appendix

(Symbols and their acronyms detected by our algorithm. Out of eight orientations of TRN, MOS, and CAP, only four have been shown.)

Active components		*#Orientations*
OPA		4
DIO		4
TRN		8
		8
NMOS/PMOS		8
FET		8
Logic gates		
INV		4
AND		4
OR		4
NAND		4
NOR		4
XOR		4
XNOR		4
Passive components		
RES		2
IND		4
CAP		8
TRF		2
Sources		
SIG		2
BAT		4