A Unified Scheme for Detecting Fundamental Curves in Binary Edge Images

浅野哲夫[1]、加藤直樹[2]、徳山豪[3]

[1]大阪電気通信大学（寝屋川市初町）
[2]神戸商科大学（神戸市西区学園西町）
[3]日本ＩＢＭ東京基礎研究所（大和市下鶴間）

２値のエッジ画像から直線、円、楕円などのディジタル曲線成分を検出することは、パターン認識における最も基本的な問題の一つである。本論文では、ある条件を満たす様々な平面曲線について、そのすべての成分を抽出するための統一的な手法を提案する。曲線族の複雑度を反映する一つの測度 d を導入し、提案するアルゴリズムが $O(n^d)$ の時間と線形記憶領域だけで実行できることを示す。ただし、n はエッジ点の個数である。

A Unified Scheme for Detecting Fundamental Curves in Binary Edge Images

Tetsuo ASANO[1]、Naoki KATOH[2]、Takeshi TOKUYAMA[3]

[1]Osaka Electro-Communication University,
[2]Kobe University of Commerce,
Tokyo Research Laboratory, IBM Japan

One of the most fundamental problems in pattern recognition is to detect digital components of planar curves such as lines, circles and ellipses in a binary edge image. In this paper we present a unified scheme for detecting all possible components of various planar curves satisfying a certain constraint. We introduce some measure d to reflect the complexity of a family of curves and show that the algorithm to be presented runs in $O(n^d)$ time and linear space where n is the number of points.

1 Introduction

One of the most fundamental problems in pattern recognition is to detect all digital line components in a given binary edge image. Although a number of methods such as the Hough Transform [4, 5, 6, 12] have been proposed for this purpose, it does not seem that any of them can guarantee to detect all line components. The algorithm based on the L_1-dual transform presented by the first two authors [3] is the first one that can guarantee perfect detection of all line components in $O(nN^2)$ time and $O(n)$ space, where n is the number of edge points (or black dots) and N the size of an input image.

This paper generalizes the idea to develop algorithms for detecting all digital components of more complex curves such as circles and ellipses. To deal with various planar curves, we consider a family of planar curves defined by linear combination of $d+1$ linearly independent polynomial expressions of x and y with d parameters. More precisely, such family is specified by an equation of the form

$$f[a_1,...,a_d](x,y) \equiv t_0(x,y) + \sum_{i=1}^{d} a_i t_i(x,y) = 0, \tag{1}$$

where each $t_i(x,y)$, $0 \leq i \leq d$ is a polynomial function of x and y, and each a_i, $1 \leq i \leq d$ is a real parameter. We assume d is a fixed constant throughout this paper. This includes most of interesting families of curves such as lines, circles, ellipses and hyperbolas in 2-D.

Figure 1 illustrates the problem to detect elliptic components. Note that a component does not necessarily form a complete ellipse but may be a part of it, and that edge points of an elliptic component do need to be on the ellipse, but have only to be close enough to it (more precisely, an edge point belongs to the elliptic component if the minimum of horizontal and vertical distances from the point to the ellipse is at most 0.5). This is a difference from the problem of recognizing ellipses.

In this paper we present a unified scheme for detecting all such digital components in a given binary edge image. The algorithm to be presented runs in $O(n^d)$ time and $O(n)$ space. One of excellent features of our algorithm is that our algorithm does not need to solve any nonlinear equation at all, and primitive operations required are only algebraic operations of bounded degrees and comparisons; thus our time complexity is attained on the algebraic decision tree model.

When specialized to ellipses (resp. circles), our algorithm computes all digital elliptic (resp. circular) components in $O(n^5)$ (resp. $O(n^3)$) time and $O(n)$ space. A number of algorithms have been proposed for detecting digital circle or ellipse components[8, 9, 11, 13], but as far as the authors know there is no known algorithm that is guaranteed to detect every such component. So, in this sense this is the first algorithm with guaranteed performance.

One of the techniques we use in our algorithm is linearization technique [7], i.e., in our approach, a family of planar curves is represented by using linear parameters as in (1). Because of linearity of parameters, the problem for detecting curves whose digital images contain digital components is naturally transformed into the one for examining a $d-$dimensional arrangement to find appropriate cells. Focusing on critical cases of such curves, the problem can be reduced to the one for searching an arrangement of lines (not "curves"!) in 2-parameter space, which enables us to apply Topological Walk algorithm [2]. Reduction to a parameter space of a different dimension could be possible. However, the reduction to the 2-parameter space is crucial in attaining our time bound. Reduction to 1-parameter space increases the time bound by $O(\log n)$ factor and no efficient algorithm is known for sweeping arrangements in $\geq 3-$dimensional space.

The third is to find

2 Definitions

Consider a 2-dimensional lattice plane of size $N \times N$:

$G = \{(x, y) \mid x, y = 0, 1, 2, ..., N-1\}$.

Given a 2-dimensional curve $f(x, y) = 0$, its digital image $Im(f)$ is defined to be a set of all lattice points in G whose horizontal or vertical distance to the curve is at most 0.5. In other words, whenever the curve intersects a lattice edge that connects two adjacent lattice points p and q, and p is closer to the intersection point than q, p is included in the digital image while q is not. If the intersection is just in the middle of p and q, both of them are included. Rosenfeld and Melter[10] discuss how to define digitalization from continuous preimage. Our definition of $Im(f)$ is reasonable from their point of view.

We are not interested in computing $Im(f)$ for a given curve $f = 0$, but in a rather inverse problem. That is, given a *binary edge image* consisting of *edge points*, which form a subset S of G, and a family of planar curves, we want to enumerate all subsets P of S whose sizes are greater than equal to a given threshold such that there exists an instance $f = 0$ belonging to the family with $P \subseteq Im(f)$.

In this paper we consider only those families of curves that can be represented by (1). A family of planar curves defined by (1) is denoted $\mathcal{F}$. When understood from the context, a member of $\mathcal{F}$ is simply written as $f(x, y) = 0$. In most of applications, we need to impose some constraints concerning parameters represented by polynomial inequalities. Let $C(\mathcal{F})$ denote the set of constraints for $\mathcal{F}$.

For a family $\mathcal{F}$ of planar curves, we define its digital component as follows.

[Definition 1] Let $\mathcal{F}$ be the one as defined above. Then, a subset P of S is a *digital component* of $\mathcal{F}$ if there exists an instance $f(x, y) = 0$ of $\mathcal{F}$ such that $P = S \cap Im(f)$.

We assume throughout this paper that the threshold of the size of a digital component that we want to detect is larger than d (i.e., the number of parameters). Notice that this assumption is natural and is satisfied in most of applications.

[Definition 2] The square region corresponding to an image G is defined as the domain D, i.e.,

$D = \{(x, y) \in R^2 \mid 0 \leq x, y \leq N-1\}$.

For an $f = 0$ in a family of $\mathcal{F}$, let us define an real image of f as

$RealIm(f) = \{(x, y) \in D \mid$ the minimum of horizontal and vertical distances from (x, y) to $f = 0$ is at most $0.5\}$.

Here we can observe the following fact: The curve $f(x, y) = 0$ may partition D into a number of regions. The sign of a value of f is identical at every point in each such region. That is, the domain is partitioned into regions of positive signs and negative signs.

Here we make an assumption on the form of a function. That is, whenever two regions are adjacent to each other, their signs must be different. Note that this assumption is not satisfied only if the function has an irreducible component which is the sum of even powers of polynomials. Thus, this assumption seems to be reasonable for practical use.

[Definition 3] Let f be an instance of a family $\mathcal{F}$ of (1). For a planar curve $f = 0$, the one resulting by shifting $f = 0$ to the right, left, above, or below by 0.5 is denoted by $f_l = 0, f_r =$

$0, f_t = 0$, and $f_b = 0$, respectively. Formally, they are defined as follows:

$$f_r(x,y) = f(x-0.5,y),\ \ f_l(x,y) = f(x+0.5,y),\ \ f_t(x,y) = f(x,y-0.5),\ \ f_b(x,y) = f(x,y+0.5).$$

For notational convenience, $f_0(x,y)$ is defined to be $f(x,y)$. Five curves $f_\gamma(x,y) = 0$, $\gamma \in \{0,l,r,t,b\}$, are referred to as the *characteristic curves* of f.

$T_r(f), T_l(f), T_t(f), T_b(f)$ defined below are called *right*, *left*, *top*, and *bottom* **territories**, respectively.

$$T_\gamma(f) = \{(x,y) \in D \mid f_\gamma(x,y) \times f(x,y) \leq 0\}, \tag{2}$$

where $\gamma \in \{r,l,t,b\}$. The union

$$T(f) = T_r(f) \cup T_l(f) \cup T_t(f) \cup T_b(f) \tag{3}$$

is simply called the *territory* of the curve $f = 0$. Clearly, $T(f) \subseteq RealIm(f)$ holds.
[Definition 4] For a given instance $f = 0$ of $\mathcal{F}$, an edge point $p = (x,y) \in S$ is called a *regular point* of $Im(f)$, if $p \in T(f)$. A subset P of S is called a *regular digital component* of $\mathcal{F}$ if there exists an instance $f(x,y) = 0$ of $\mathcal{F}$ such that $P = S \cap T(f)$. A curve $f(x,y) = 0$ is *regular* if $T(f) = RealIm(f)$, and is *irregular* otherwise.

Let $\mathcal{F}$ be a family of planar curves with a set of constraints $C(\mathcal{F})$. If there is a set of parameters such that for any values of them we can adjust other parameters so as to satisfy $C(\mathcal{F})$, the set of parameters is called a *free parameter set*. We are interested in a maximal free parameter set and the size of the maximal free parameter set is denoted by $free(\mathcal{F})$.

3 Algorithm for Detecting All Digital Components

The problem of detecting all regular digital components for planar curves is formally stated as follows.

[Problem 1] Given a binary edge image consisting of n edge points, a family $\mathcal{F}$ of curves with the constraint set $C(\mathcal{F})$ and a threshold k on the size of a component, enumerate all possible regular digital components of $\mathcal{F}$ whose sizes are at least k. Here, $k \geq d$ is assumed.

This section presents an $O(n^d)$ time and $O(n)$ space algorithm for this problem.

For an edge point $p = (x,y) \in S$, the condition with resect to parameters that there exists an instance $f \in \mathcal{F}$ such that $p \in T(f)$ is expressed by

$$\bigcup_{\gamma \in \{r,l,t,b\}} \{(a_1,\ldots,a_d) \mid (f_\gamma(x,y) \geq 0 \wedge f(x,y) \leq 0) \vee (f_\gamma(x,y) \leq 0 \wedge f(x,y) \geq 0)\} \tag{4}$$

from (3). Thus, the region in d-dimensional parameter space satisfying (4) forms a union of polyhedra.

Thus, the following theorem is immediate from this observation.

Theorem 1 *Let $\mathcal{F}$ be any family of planar curves and P be an arbitrary regular digital component of $\mathcal{F}$, and assume that the size of P is greater than d. Then, there always exists an instance f of $\mathcal{F}$ with $P \subseteq T(f)$ such that for some r with $0 \leq r \leq free(\mathcal{F})$, (i) r edge points of P lie on one of its five characteristic curves, and (ii) $free(\mathcal{F}) - r$ parameters can be arbitrarily fixed.*

Based on this theorem, our algorithm is constructed. The algorithm to be presented below can be applied to a family of curves with d parameters such that $free(\mathcal{F}) \geq d-2$.

First of all, the algorithm chooses a set of $d-2$ free parameters, denoted A, among $free(\mathcal{F})$ parameters. Without loss of generality, let $A = \{a_1, a_2, \ldots, a_{d-2}\}$. Let $A' = \{a_{d-1}, a_d\}$.

In the algorithm we do the following by choosing every set P of $d-2$ edge points. Let us assume that the chosen points are $p_1 = (x_1, y_1), \ldots, p_{d-2} = (x_{d-2}, y_{d-2})$. For each point $p_i \in P$, we choose one of five characteristic curves and associate it with p_i. Let $f_{\gamma_i} = 0$ be the characteristic curve associated with p_i. Using Theorem 1, we test whether such $d-2$ points of P together with the associated characteristic curves satisfy the assertion of Theorem 1. It is done as follows. Suppose all these $d-2$ points lie on the associated characteristic curves, define

$$t_i^0(x,y) = t_i(x,y), t_i^r(x,y) = t_i(x-0.5,y), t_i^l(x,y) = t_i(x+0.5,y),$$
$$t_i^t(x,y) = t_i(x,y-0.5), t_i^b(x,y) = t_i(x,y+0.5).$$

for each i with $0 \leq i \leq d$, and let T be a $(d-2) \times d$ matrix whose (i,j) component is $t_j^{\gamma_i}(x_i, y_i)$. Then we obtain the following $d-2$ linear equations on d parameters:

$$(T', T'')a = b \tag{5}$$

Here, T' and T'' are $(d-2) \times (d-2)$ and $(d-2) \times 2$ submatrices, respectively, such that $T = (T', T'')$, $a = (a_1, a_2, \ldots, a_d)$ is d-dimensional column vector, and b is $(d-2)$-dimensional column vector whose i-th element is $-t_0^{\gamma_i}(x_i, y_i)$.

Next we try to solve the equation (5) with respect to $a_1, a_2, \ldots, a_{d-2}$. This is done by testing whether $det(T') \neq 0$, which can be accomplished by applying the conventional Gaussian elimination method.

Case 1. $det(T') \neq 0$. In this case, we can express every parameter of A as a linear combination of a_{d-1} and a_d, which implies that we can choose a set of parameters A such that all $d-2$ points lie on the associated characteristic curves.

Substituting every parameter of A expressed as a linear combination of a_{d-1} and a_d into (4), the condition concerning a parameter set $(a_1, \ldots, a_d)$ that determines the existence of f with $p \in T(f)$ for each edge point p of $S-P$ can be expressed by a (possibly not connected) polygonal region, denoted $R(p)$, in (a_{d-1}, a_d) plane (see Figure 4). Notice that the edegs of $R(p)$ forms an arrangement of five lines in the (a_{d-1}, a_d) plane, since $f_\gamma(x,y) \geq 0$ (or ≤ 0) for an edge point $p = (x,y)$ and for some $\gamma \in \{0, r, l, t, b\}$ corresponds to a halfplane in the plane.

In this way we have $n-d+2$ such polygonal regions in (a_{d-1}, a_d) plane. The set of edges of these regions makes an arrangement of $O(n-d+2)$ lines in the plane, and subdivides the plane into $O(n^2)$ cells. Then, all we need to do is to enumerate all the cells at which k or more such regions intersect. As will be explained in details later, this task will be done in $O(n^2)$ time using linear space by applying the Topological Walk algorithm [2].

Case 2. $det(T') = 0$. In this case, applying the Gaussian elimination method with slight modifications, we determine r independent row vectors of T', where $r = rank(T')$, and express each of the remaining $d-2-r$ row vectors of T' as a linear combination of such r vectors.

In terms of our problem, we could lay r points of P corresponding to r independent row vectors of T' on the associated characteristic curves. Let P' be the set of such points and P'' the set of the other $d-2-r$ points. However, it is not clear at this point whether there exists an instance f such that $T(f)$ contains P. We shall perform this test as follows. $r = rank(T')$ and (5) imply that we can express r parameters as a linear combination of a_{d-1} and a_d by fixing the rest of $d-2-r$ parameters to arbitrary values. Because of $rank(T') = r$, substituting these expressions into $d-2-r$ linear equations of (5) corresponding to P'' results in linear equations of only two parameters a_{d-1} and a_d. It is probable that such $d-2-r$ linear equations of a_{d-1} and a_d cannot be simultaneously satisfied for any choice of a_{d-1} and a_d. However, for our purpose, we only have to determine whether there exists f such that the points of P'' lie in $T(f)$. Since the condition concerning parameters that determines the existence of such f can be expressed by a polygonal region $R(p)$ for $p \in P''$, this test can also be done in $O((d-2-r)^2)$ time by the same manner as in Case 1 using the topological walk algorithm in (a_{d-1}, a_d) plane.

If no such f exists for some point of P'', we discard the currently chosen $(d-2)$-tuple of points and choose the next one. Otherwise, after expressing all r parameters as linear combinations of a_{d-1} and a_d by fixing the rest of $d-2-r$ parameters to arbitrary values, we can again apply the topological walk algorithm as in Case 1 to accomplish our task.

As was seen in both of Cases 1 and 2, we have to solve the following problem. It can be solved efficiently using the topological walk algorithm.

[Problem 2] Given some integer k and a collection of polygonal regions each defined by some constant number of edges, enumerate all those portions at which k or more such regions intersect and report coordinates of a point in each such portion.

We need some assumptions:

(1) Each polygonal edge is directed from left to right, or upward if it is vertical), and each edge e is associated with the two lists RIGHT(e) and LEFT(e), where RIGHT(e) (LEFT(e), respectively) is the list of names of polygonal regions which have e at their boundaries and lie to the right (left, respectively) of e.

(2) The lengths of the lists RIGHT and LEFT are $O(1)$.

(3) Although two or more edges may lie on the same line, the maximum number of edges lying on a line is bounded by some constant.

With these assumptions we can describe an algorithm for solving the Problem 2 above. The key idea is the Topological Walk. First of all we extend each polygonal edge in both directions to define a line. After defining all the lines, for each line we create a list of all edges included in it. Note that the assumption (3) above guarantees that the length of the list is $O(1)$. In addition, we make the lists RIGHT(e) and LEFT(e) for each edge e.

Now, we have $O(n)$ lines, which partition the parameter plane into $O(n^2)$ cells. What is required here is to detect all those cells at which at least k regions intersect. Topological Walk is an algorithm for visiting cells in an arrangement of lines in time proportional to the number of cells. It starts with one cell. Then, it iteratively visits cells neighboring to the previous cells until it exhausts all the cells. Some cells may be visited more than once, but the number of cells visited in total remains $O(n)$.

For the starting cell, we find a set of polygonal regions which intersect it. This is done in $O(n)$ time. Then, each time we move to a neighboring cell we need to update the set of polygonal regions which intersect the cell. If we step over an edge e to move from a cell C to its neighboring one C' which lies to the left of e, we can obtain the set for C' just by deleting the elements of RIGHT(e) and adding ones of LEFT(e). Since we know the names of lines containing the edge e and the assumption (3) guarantees that only constant number of edges are included in the list associated with the line, only constant time is required to retrieve the information of RIGHT and LEFT. If C' lies to the right of e, we delete LEFT(e) and add RIGHT(e). Since the lengths of RIGHT and LEFT are bounded by some constant, this operation is done in constant time. Thus, the total time needed is $O(n^2)$.

Back to Problem 1, we examine all possible $(d-2)$–tuples of points. We have $O(n^{d-2})$

different ways of choosing $d-2$ points. For each of chosen $d-2$ points, we choose one of five different characteristic curves. Thus, in total there are $O(5^{d-2})$ different ways for associating each point with a characteristic curve.

For each such way we try to lay those $d-2$ edge points on characteristic curves specified as many as possible. As was explained above, this can be done by applying the Gaussian elimination method, which requires $O(d^3)$ time. Therefore, in total the algorithm requires $O(d^3 5^d n^d)$ time and $O(d^2+n)$ space.

Theorem 2 *The algorithm above correctly detects all possible regular digital components for a family of curves $\mathcal{F}$ in $O(d^3 5^d n^d)$ time and $O(d^2+n)$ space.*

4 Conclusions

In this paper we have presented a unified scheme for detecting digital components of a family of curves. It seems to give optimal algorithms in most cases. A key to this success is due to application of algorithmic techniques developed in Computational Geometry such as Topological Walk and Linearization technique. Although the theory described here is of general nature, it is also practically useful. In particular, as far as the authors know, this is the first linear-space algorithm for detecting circle and ellipse components.

References

[1] T. Akutsu, Y. Aoki, S. Hasegawa, H. Imai, and T. Tokuyama, The Sum of Smaller Endpoint Degree over Edges of Graphs and Its Applications to Geometric Problems, *Proc. 3rd CCCG* (1991) pp.145-148.

[2] T. Asano, L. Guibas, and T. Tokuyama, Walking in an Arrangement Topologically, to appear in *Int. J. of Comput. Geom. and Appl.*

[3] T. Asano and N. Katoh, Number Theory Helps Line Detection in Digital Images, to appear in *Proc. ISAAC'93.*

[4] D.H. Ballard, Generalizing the Hough Transform to Detect Arbitrary Shapes, *Pattern Recognition*, 13, 2 (1981), pp. 111-122.

[5] C.M. Brown, Inherent Bias and Noise in the Hough Transform, *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-5, No.5 (1983), pp. 493-505.

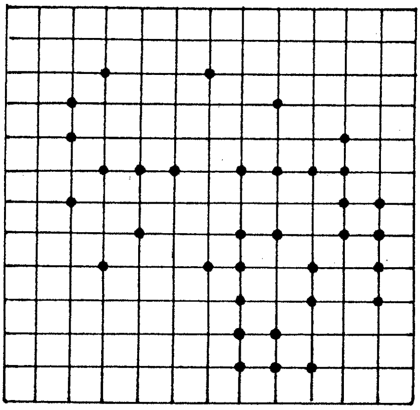
[6] R. O. Duda and P.E. Hart. Use of the Hough Transformation to Detect Lines and Curves in Pictures, *Comm. of the ACM*, 15, January 1972, pp.11-15.

[7] K. Imai, S. Sumino, and H. Imai, Minimax Geometric Fitting of Two Corresponding Sets of Points, *Proc. 5th Symposium on Computational Geometry*, (1989), pp. 276-282.

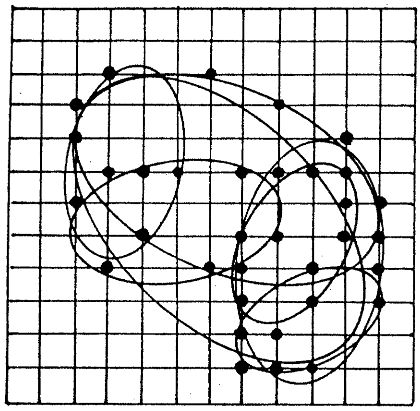
[8] C. Kimme, D.H. Ballard, and J. Sklansky, Finding circles by an array of accumulators, *Comm. ACM*, 18 (1975), pp. 120-122.

[9] H. Maitre, Contribution to the Prediction of Performance of the Hough Transform, *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-8, No.5 (1986), pp. 669-674.

[10] A. Rosenfeld and R.A. Melter, Digital Geometry, Tech. Report, CAR-TR-323, Center for Automatic Research, University of Maryland, 1987.



(a) Input binary edge image.



(b) Elliptic components to be detected.

Figure 1: Detection of elliptic components.

[11] P. Sauer, On the Recognition of Digital Circles in Linear Time, *Computational Geometry: Theory and Applcations*, 2 (1993), pp. 287-302.

[12] J. Sklansky, On the Hough technique for curve detection, *IEEE Trans. Comp.*, C-27, 10 (1978), pp. 923-926.

[13] S. Tsuji and F. Matsumoto, Detection of ellipses by a modified Hough transformation, *IEEE Trans. Comp.*, C-27, No. 8 (1978), pp. 777-781.