1998 International Computer Symposium
Workshop on Image Processing and Character Recognition
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

DETECTING K LINE SEGMENTS IN AN IMAGE - A NEW IMPLEMENTATION FOR HOUGH TRANSFORM

Yu-Tai Ching
Department of Computer and Information Science
National Chiao Tung University
Hsinchu, Taiwan R. O. C.

Abstract

The conventional Hough transform is a technique for detecting line segments in an image. Conventional Hough Transform transforms image points to lines in the parameter space. If there are collinear image points, the lines transformed from the points intersect at a point. To find out the intersection is generally carried out through the "voting method" which partitions the parameter space into squared meshes. A problems with the voting method is to determine the resolution required for partitioning the parameter space. In this paper, we present a solution to this problem. We propose to transform an image point to a belt. We show that a very good approximation for the width of the belt can be derived. An algorithm for detecting line segments based on the transformation is designed. This algorithm takes $O(n \log n)$ time where n is the number of image points. Since a tight enough bound is available, the proposed algorithm can be applied to detect k line segments in an image where k is not known in advance. Experiments show that the proposed algorithm is very robust in almost every case.

1 Introduction

The *Hough Transform* (HT) is a technique in Pattern Recognition which has been applied to identify shapes of vary kinds in an image[1, 2, 3, 4, 5]. The basic idea of HT transforms points in the *image space* to geometric objects in another *parameter space*. The shape in the image space is then identified by finding the common intersection of many geometric objects in the parameter space.

The conventional Hough Transform for identifying line segment in an image is generally carried out though the following procedures. An image is preprocessed by thresholding, edge detection, or thinning in order to produce a binary image which could possibly contain line pattern of 1-pixel width. Each image point (i, j) is then transformed to a line $y - ix - j = 0$ in the parameter space. If there is a set, S, of collinear image points, the set of lines transformed from S have common intersection.

By applying this transformation, a line having slope close to ∞ (a vertical line) is difficult to detect since the set of points on a vertical line are transformed to a set of parallel lines. This case is known as the *degenerate case* which can be eliminated by applying another transformation.

A line in the xy-plane can also be specified by another two parameters, θ, the orientation of the line and ρ, the distance between the line to the origin of the xy-plane. A point in the xy-plane can be transformed to a sinusoidal curve in (θ, ρ)-parameter space by the mapping

$$\rho = x \cos(\theta) + y \sin(\theta).$$

Still, the set of curves intersect at a point if they are transformed from a set of collinear points. This transformation eliminates the degenerate case stated previously but it is less efficient because the need of computing the trigonometry functions. One way to improve the efficiency is to use a lookup table.

Regardless the kinds of transformation applied, implementation of Hough Transform

needs to calculate the intersection of many geometric objects. A typical approach is a discrete approach called "voting". The parameter space is first partitioned into squared meshes with a "proper resolution". Each square holds a counter which counts the number of lines passing through it. The square having the largest vote is the intersection point. A problem with this approach is to determine the "proper resolution". Underestimate the resolution reduces accuracy. High resolution increases the accuracy but sacrifices both the time and space efficiency. The real problem is that we do not know the required resolution. Another problem is that the computation time required by the "voting" approach does not totally depend on the number of image points. It also depends on the resolution.

Some methods have been proposed to solve the resolution issue. These methods were designed based on the observation that "high resolution is required only at the place where there are many intersection points". Dynamic tree structures, which always maintained a small amount of nodes, was designed to keep track such hot spot regions[6]. Another strategy was to iteratively focus at the place having many intersection points from coarse to fine resolution[7].

In this paper, we study the resolution issue and propose a line segment detection algorithm. The proposed algorithm is based on a fact that a line segment in an image has width. When we say that there is a set of points collinear to L, we actually mean that each point in the set has distance less than ϵ to L. Since these points are not on a mathematical line, the transformation of the points does not intersect at a point. Thus, we propose to transforms a point to a belt. We show that the width of the belt is a function of the width of the line segment in an image. To find out the common intersection of many lines becomes to determine the intersection of many belts. This can be solved in $O(n \log n)$ time where n is the number of image points. Since we can give a good approximation for the width of the belt, the proposed algorithm can be extended to detect k line segments where k is not known in advance.

In the next section, we describe the transformation that we shall use in this paper. The details of the proposed algorithm will be presented in Section 3. Section 4 contains the experimental results. The conclusion is in Section 5.

2 Hough Transform and Geometric Duality

Hough Transform is known as "Geometric Duality" in the area of Computational Geometry[8]. There are several dual transforms between geometric objects studied in Computational Geometry. A general paradigm is to transform an 1-flat (a point) to a $(k-1)$-flat and vise versa in a k-dimensional space. The 1-flat and the $(k-1)$-flat are *duals* to each other. For examples, point and line (point and plane) are duals to each other in 2-D (3-D) space.

One of such transformations transforms a point (a, b) to a line $ax + by = 1$ and vise versa in an xy-plane. This dual relationship is a transformation between a point and a line through a unit circle centered at $O = (0, 0)$. Consider a point p and its dual $D(p)$ as shown in Figure 1. $D(p)$ is a line perpendicular to $\overline{O,p}$ and $d(O, D(p)) = 1/(d(O, p))$ where $d(P_1, P_2)$ is the distance between two geometric objects P_1 and P_2.

This dual transformation has the following properties.

1. $D(D(P)) = P$, P is either a point or a line.
2. Let p and q be two points. The dual of the intersection of $D(p)$ and $D(q)$ is the line $\overline{p,q}$.
3. The duals of all the points on the line $\overline{p,q}$ pass through $D(\overline{p,q})$.

There are also degenerate cases in this transformation. The origin of the unit circle, O, does not have a line as its dual. Lines passing through the origin of the unit circle do not have duals either.

In this paper, we modify the dual transformation so that a point, p, is transformed to a

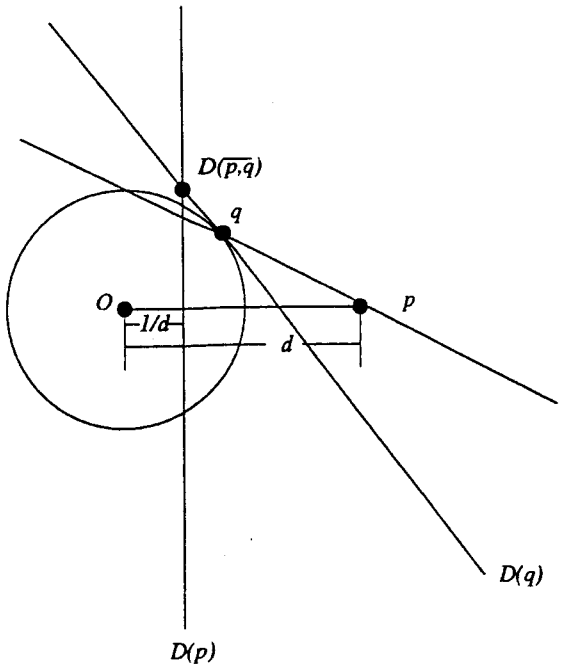

Figure 1: p is transformed to $D(p)$ through a unit circle centered O. $d(O,p) = d$ and $d(O, D(p)) = 1/d$. Duals of all the points on $\overline{p,q}$ intersect at $D(\overline{p,q})$.

belt. In order to distinguish the dual in the original transformation (a line) and the dual in the modified transformation (a belt), we use $D(p)$ to denote the line and $D'(p)$ to denoted the belt. $D'(p)$ can be specified by a pair of parallel lines which are on the both sides of $D(p)$.

3 The Proposed Algorithm

In this section, we show that the width of the transformed belt is a function of the width of a line segment in an image. The equations for the parallel lines representing the belt will be derived from the width of the belt. The algorithm for detecting line segments in an images will also be presented in this section.

3.1 The Width of the Line Segment and the Width of the Belt

Suppose that a point p is on line L if $d(p, L) \leq \epsilon$, i.e., the width of a line segment is 2ϵ. Let the width of the belt $D'(p)$ be 2δ. The following Theorem gives the relationship between ϵ and

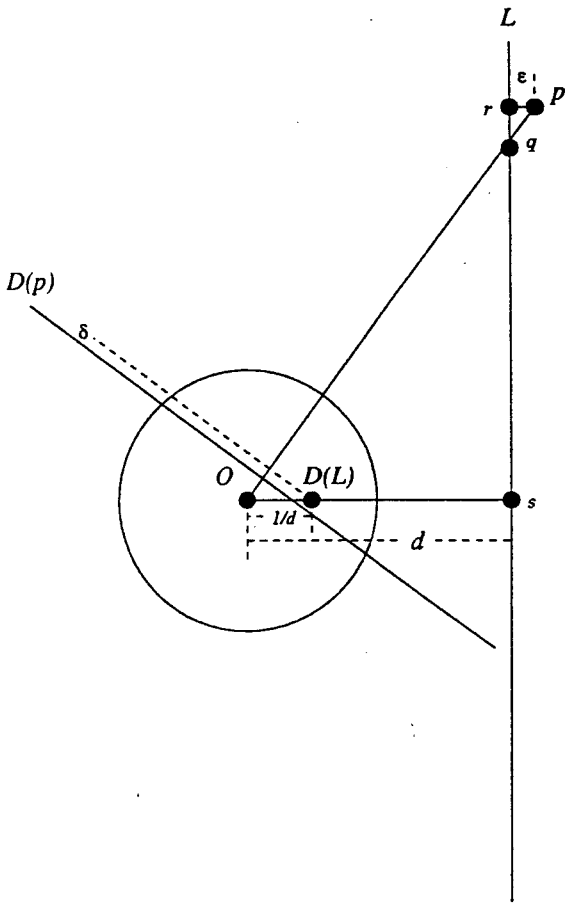

Figure 2: Since p is not on L and $d(p, L) = \epsilon$, the dual of p does not pass through $D(L)$ but δ away from $D(L)$.

δ.

Theorem Suppose that a point p is on a line segment L if $d(p, L) \leq \epsilon$. A point on L is transformed to a belt with width 2δ. Then we have

$$\delta = \frac{\epsilon}{d_2 d_1}, \quad (1)$$

where d_1 and d_2 are $d(L, O)$ and $d(p, O)$ respectively.

Proof We assume that p and O are on the different sides of L as shown in Figure 2. Let q be the intersection between $\overline{O,p}$ and L. Since $\overline{p,r}$ is parallel to $\overline{O,s}$, we have

$$\frac{\epsilon}{d_1} = \frac{d(p,q)}{d_2 - d(p,q)}. \quad (2)$$

Since p is not on L, $D(p)$ does not pass through $D(L)$. Let $d(D(p), D(L)) = \delta$. We have

$$\frac{1}{d_2 - d(p,q)} - \delta = \frac{1}{d_2}. \quad (3)$$

From Equation (2), we have

$$\epsilon d_2 - \epsilon d(p,q) = d_1 d(p,q),$$

and thus

$$d(p,q) = \frac{\epsilon d_2}{d_1 + \epsilon}.$$

From Equation (3),

$$\begin{aligned} \delta &= \frac{1}{d_2 - d(p,q)} - \frac{1}{d_2} \\ &= \frac{1}{d_2 - \frac{\epsilon d_2}{d+\epsilon}} - \frac{1}{d_2} \\ &= \frac{d_1 + \epsilon}{d_2 d_1} - \frac{d_1}{d_2 d_1} \\ &= \frac{\epsilon}{d_2 d_1}. \end{aligned}$$

If p and O are on the same side of L, Equations (2) and (3) become

$$\frac{\epsilon}{d_1} = \frac{d(p,q)}{d_2 + d(p,q)}$$

and

$$\delta = \frac{1}{d_2} - \frac{1}{d_2 + \frac{\epsilon d_2}{d_1 - \epsilon}}.$$

We still have

$$\delta = \frac{\epsilon}{d_2 d_1}.$$

□

3.2 Equations for $D'(p)$

From the above theorem, if the width of a line segment is 2ϵ, the width of a belt will be be $2\epsilon/(d_2 d_1)$. If $D(p)$ is $ax + by = 1$, $D'(p)$ are parallel lines on both sides of $D(p)$ and $\epsilon/(d_2 d_1)$ away from $D(p)$. Let

$$\begin{aligned} ax + by &= C' \quad \text{and} \\ ax + by &= C'' \end{aligned}$$

be these two lines. We now show that $C' = 1 + \epsilon/d_1$ and $C'' = 1 - \epsilon/d_1$.

Consider a point $p = (a, b)$ and its dual $D(p)$ $= ax + by = 1$. $(0, 1/b)$ is a point on $D(p)$ and both $N' = (a, b)$ and $N'' = (-a, -b)$ are normal vectors for $D(p)$. The normalized N' and N'' are

$$N' = (\frac{a}{\sqrt{a^2+b^2}}, \frac{b}{\sqrt{a^2+b^2}})$$

and

$$N'' = (-\frac{a}{\sqrt{a^2+b^2}}, -\frac{b}{\sqrt{a^2+b^2}}).$$

$D'(p)$ are parallel lines which must respectively pass through the two points $N'\delta + (0, 1/b)$ and $N''\delta + (0, 1/b)$. These two points are

$$N'\delta + (0, \frac{1}{b}) =$$

$$(\frac{\epsilon a}{d_1 d_2 \sqrt{a^2+b^2}}, \frac{\epsilon b}{d_1 d_2 \sqrt{a^2+b^2}} + \frac{1}{b}) \quad \text{and}$$

$$N''\delta + (0, \frac{1}{b}) =$$

$$(-\frac{\epsilon a}{d_1 d_2 \sqrt{a^2+b^2}}, -\frac{\epsilon b}{d_1 d_2 \sqrt{a^2+b^2}} + \frac{1}{b}).$$

The pair of parallel lines of $D'(p)$ must respectively pass through these two points. We then have

$$a(\frac{\epsilon a}{d_1 d_2 \sqrt{a^2+b^2}}) + b(\frac{\epsilon b}{d_1 d_2 \sqrt{a^2+b^2}} + \frac{1}{b}) = C' \quad \text{and}$$

$$a(-\frac{\epsilon a}{d_1 d_2 \sqrt{a^2+b^2}}) + b(-\frac{\epsilon b}{d_1 d_2 \sqrt{a^2+b^2}} + \frac{1}{b}) = C''.$$

Since $d_2 = \sqrt{a^2+b^2}$, C' and C'' are respectively

$$C' = 1 + \frac{\epsilon}{d_1} \quad \text{and} \quad (4)$$

$$C'' = 1 - \frac{\epsilon}{d_1}. \quad (5)$$

3.3 The Algorithm

From Equations (4) and (5), C' and C'' depend on d_1 and ϵ. To get an accurate estimation for ϵ is not a straightforward task. A much more difficult problem is to determine d_1 since we do not know where the line is. Fortunately, the proposed algorithm works very well with rough but reasonable estimation for ϵ and d_1.

We now derive an estimation for d_1. We first restrict our discussion to detect a line segment in an image with negative slope. Suppose that the size of an image is W by H. If we centered the unit circle at (0,0), then d_1 is ranged from 0 to $\sqrt{W^2+H^2}$. That means $1/d_1$ ranged over $1/\sqrt{W^2+H^2}$ to ∞. In this case, an estimation for d_1 could cause significant error. Let $T = \max(H, W)$. If we translate all the image points (i, j) to $(i+cT, j+cT)$ where c is a positive constant. Then for any line with negative slope, we have $cT \leq d_1 \leq \sqrt{(cT+W)^2+(cT+H)^2} \leq \sqrt{2}(c+1)T$, or

$$\frac{1}{\sqrt{2}(c+1)T} \leq \frac{1}{d_1} \leq \frac{1}{cT}.$$

Let

$$R = \frac{c}{\sqrt{2}(c+1)}.$$

If the estimation for d_1 is $\sqrt{2}(c+1)T$. Since the actual distance is ranged over cT to $\sqrt{2}(c+1)T$. In the worst case, we could overestimate d_1 for at most $(1/R)$ times the actual distance. Note that $1/R$ approaches $\sqrt{2}$ as c increasing. When c is 9, $1/R$ is very close to $\sqrt{2}$, and double precision variable in any programming language can provide sufficient numerical precision for calculating $1/(\sqrt{2}(c+1)T)$. Since we could overestimate d_1, we are allowed to overestimate ϵ. According to the experiments, the proposed algorithm is not sensitive to ϵ. We can give a rough estimation for ϵ which is allowed several pixels greater than the actual ϵ. To detect a line segment having positive slope, we center the unit circle at $(0, H)$ and let an image point (i, j) to be $(i+cT, -j-cT)$.

We now present the line segment detection algorithm. Let S be a set of image points. Assume that the slope of a line segment is negative. The following pseudo code detects a line segment containing points in S.

1. Let the center of the unit circle be (0,0).
2. Translate each image point (i, j) to $(i+cT, j+cT)$.
3. Randomly choose m points from the image points.
4. For each chosen point p_i, $i = 1, \ldots, m$, we do the following
   (a) Transform p_i to $D(p_i)$.
   (b) Transform all the other points q to $D'(q)$.
   (c) Find the interval on $D(p_i)$ intersected by the largest number of $D'(q)$. The clique number is p_i^{clique} and the mid-point of the interval is p_i^{center}.
5. Let P be the p_i^{center} whose associated p_i^{clique} is the largest among the m points.
6. $D(P)$ is the line containing the line segment in the image.
7. All the image points have distance less than ϵ to $D(P)$ are the points on the line segment.

Step 4c in the above pseudo code is carried out through the following procedure.

1. Find the intersections between $D(p_i)$ and the pair of lines of $D'(q)$ for all q.
2. Sort the intersection points along $D(p_i)$.
3. Each interval on $D(p_i)$ between a pair of consecutive intersection points is associated with a counter. We scan the intersection points from left to right. If a point is intersected by a left-line of $D'(q)$, the counter associated with the interval to the right of the point is increased by one. The counter is decreased by one if the point is on the right-line of $D'(q)$.
4. The largest number among all the counters is p_i^{clique}

Since we do not know the slope of the line segment in an image. We actually execute the algorithm stated above twice. The first iteration is as shown in the above. In the second

iteration, we set the center of the unit circle to $(0, H)$ and transform each image point to $(i+cT, -j-cT)$. Finally, we select $P = p_i^{\text{centre}}$ where p_i^{clique} is the largest obtained from the two iterations.

In order to detect k line segments in an image where k is not known in advance. We apply the algorithm stated above iteratively. Each time we detect a line segment and remove the image points on the line segment. The algorithm iterates until there are only sparse points left in the image.

4 Experimental Results

Experiment has been done on many data sets. Each image is preprocessed by using a Sobel filter followed by thresholding for boundary detection.

Figure 3 shows a set of street-map like image. The streets are perfect straight lines. Estimation for the width of the line segments was 3.0. The proposed algorithm could detect the line segments well even when there are lines having similar equations in this test data.

In Figure 4, the image contains line segments which are the boundaries of different width lines. There are many parallel lines in the image. When ϵ was set to 2, these parallel lines could be accurately identified.

Figure 5 shows a set of free hand drawing line segments. Since the line segments were not perfect straight lines, the estimation for the width of the line segments was set to 6.

Image in Figure 6, was obtained by photo copying matches. The estimation for the width of the matches was 4.0. In this experiment, some matches were fail to be detected since there were too few image points on these matches after the boundary detection step. The proposed algorithm regarded those line segments as noise.

Image in Figure 7 contains a circle and some line segments. The proposed algorithm could easily identify the line segments. A very interested result is that the algorithm finds eight tangent lines around the circle so that a polygon with eight side can well approximate the circle.

According to the experiment, the proposed algorithm is very roboust. It is not very sensitive with the estimation for *epsilon* and *delta*. The computing time depends on the number of points processed and the line segments detected.

e set a high threshold (80 out of 255) after Sobel filter for

5 Conclusion and Discussion

In this paper, we solve the problem with the resolution issue for the conventional Hough Transform. We show that it is not necessary to use a discrete approach (the voting method) for computing the intersection of many lines in the parameter space.

In the step 4 of the algorithm, we randomly selected m points from the image points. If the selected point is close to the center of a line the clique number p^{clique} obtained tends to be larger than the others. More points selected means we have better chance to find a point close to the center of the line. In our experiment, m=15 worked very well for all the test cases. In many cases, m=10 worked fine. All the results shown in this paper were obtained when m=10.

According to the experiments, the proposed algorithm is very robust. For example, the parameter m stated above and the threshold for eliminating sparse points are not sensitive to the computation results. Actually, the only important parameter is ϵ. And we only need a reasonable estimation for ϵ that can always achieve good results.

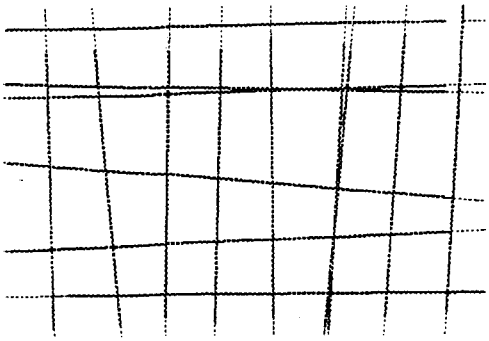

Figure 3: This data set consists of a street-map like straight lines. ϵ was 3.

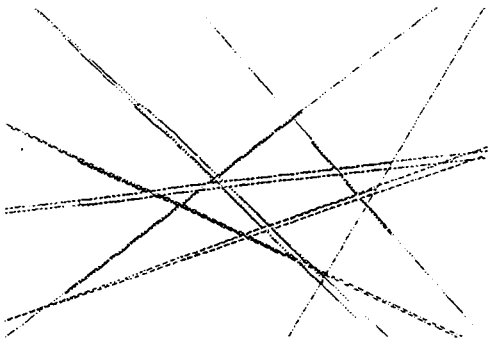

Figure 4: These set of lines are the boundaries of lines width different width. ϵ was 2.0.

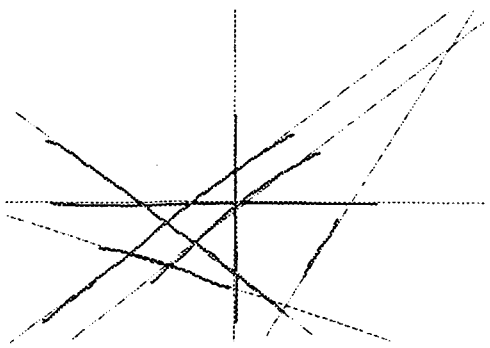

Figure 5: A set of free hand drawing lines. ϵ was 9.0.

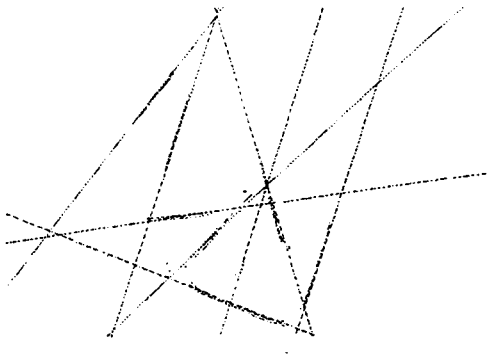

Figure 6: An image obtained by Xerox copying some matches. ϵ was 6.

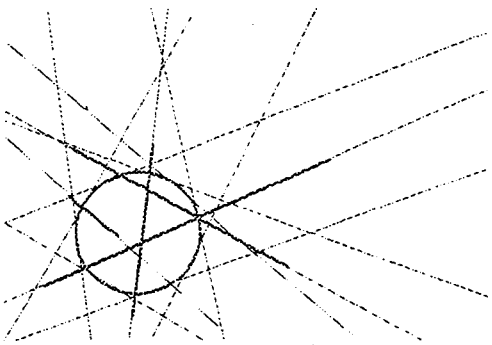

Figure 7: An image consists of line some line segments and a circle. ϵ was 5.

References

[1] P.V.C. Hough, "Method and Means for Recognizing Complex Pattern," U.S. Pattern 06954, 1962.

[2] R.O. Duda, and P. E. Hart, "Use the Hough Transform to Detect Lines and Curves in Pictures, *CACM*, 15, pp. 11-15, 1972.

[3] J. Illingworthe and J. Kittler, "A survey of the Hough Transform," *Computer Vision, Graphics, and Image Processing*, 44. 87-116 (1988).

[4] L.M. Murphy, Linear feature detection and enhancement in noisy images via Random transform, PRL 4. 279-284

[5] R. O. Lo and W. H. Tsai, "Gray-Scale Hough Transform for Thick Line Detection in Gray-scale Images," *Pattern Recognition*, Vol. 28, No. 5, pp. 647-661, 1995.

[6] L. O'Rourke and K.R. Sloan, "Dynamic Quantization: Two adaptive data structures for multidimensional space," *IEEE T-PAMI* 6, 266-279

[7] J. Illingworth and J. Kittler, The adaptive Hough Transform, IEEE T-PAMI 9, No. 5, 679-690

[8] Lee, D. T. and Ching, Y. T., 1985, "The Power of Geometric Duality Revisited," Information Processing Letter, 21, pp. 117-121.