Determining Object Orientations from Run Length Encodings

Visvanathan Ramesh
Roger W. Ehrich

TR 87-18

DETERMINING OBJECT ORIENTATIONS FROM RUN LENGTH ENCODINGS

by

Visvanathan Ramesh
Roger W. Ehrich

CS Technical Report TR 87-18

Departments of Electrical Engineering and Computer Science
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061

June 21, 1987

10th IASTED Symposium
on Robotics and Automation
Lugano, CH, June 1987

Determining Object Orientations from Run Length Encodings

Visvanathan Ramesh
Roger W. Ehrich

Departments of Electrical Engineering and Computer Science
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061, USA

ABSTRACT

Run length codes are widely used for image compression, and efficient algorithms have been devised for identifying objects and calculating geometric features directly from these codes. However, if the image objects are rotated it can be difficult to determine their orientation and position so that they can be grasped by manipulators. This paper describes a method for structural determination of object orientation directly from the run length codes of successive image scan lines.

An algorithm is described that makes use of the equations of object boundary segments to form hypotheses about object orientations that are refined as scanning progresses. 2-dimensional polygonal objects are discussed, and it is assumed that objects do not touch or overlap, although the algorithm could be extended to include those situations.

1 INTRODUCTION

Binary vision systems have often been used for industrial robotics applications to determine the identities, locations, and orientations of component parts. Perhaps the best known early industrial binary vision system is the Stanford Vision Module (Agin and Duda, 1975 and Gleason and Agin, 1979), which was marketed as the VS-100 system by Machine Intelligence Corporation. In 1980 Automatix, Inc. combined the SRI-developed technology with modern camera and processor features to produce the Autovision systems it markets today (Reinhold and VanderBrug, 1980). In Europe, the VS-100 was followed by faster and more capable systems, such as MODSYS (Foith, 1982), which was designed at the Fraunhofer Institute in Karlsruhe, Germany and marketed by Robert Bosch. All these systems rely on run length encoding of video information as a means of data compression.

Run length coding has been used extensively for compressing image data, and recently there has been interest in the design of pipelined vision processors that read and interpret the run length coded outputs of line scan cameras as objects move by. To achieve maximum speed it is desirable to maximize the processing that is done during image acquisition. For example, Shapiro, Lee, and Haralick (1985) have proposed a VLSI architecture for connected components labeling so that the principal image regions can be generated in hardware rather than in software to maximize speed. This paper discusses techniques for identifying objects as they are scanned, based upon structural information supplied by the CAD/CAM system with which they were designed.

The support of this work under Grant INF-86-006 from the Virginia Center for Innovative Technology is gratefully acknowledged.

Binary vision processors are usually equipped with noise cleaning preprocessors to remove small unconnected regions and fill small object holes. Typically this is done using morphological operators such as shrink and expand; pipelining requires that these operations be done sequentially. In this paper, however, it is assumed that a cleaned run length image encoding is available in real time but possibly with constant delay while the preprocessing takes place. The goal is to study the feasibility of identifying scene objects on the fly based on object boundary relationships.

It should be pointed out that numerous run length codes are in common use. Run length codes have two principal forms — $(i, l(i))$ and $(i, x(i))$, where i is an image intensity, $l(i)$ is the length of a run with intensity i, and $x(i)$ is the horizontal coordinate of a transition to intensity i. The first form achieves much greater compression when there are many regions but suffers from the fact that all the preceding runs must be summed to obtain the absolute horizontal coordinate for the beginning of a specific run. Since the vision processors under discussion are designed for a minimal number of regions, the second form is more convenient and is the one that is almost always used.

The SRI algorithm obtains statistical information about object surfaces from training data, whereas this work follows more recent attempts to obtain the needed structural information from a CAD-CAM system (Henderson, et al., 1986). Specifically, this paper proposes an algorithm which makes hypotheses about an object's orientation and then refines the hypotheses as the scanning progresses. The basic idea used in this paper is that one can obtain the relationship between run lengths of successive scan lines for a particular orientation using the equations of an object's edges. It is assumed that objects are 2D polygons with no holes. It is also assumed that no two objects touch or overlap. This makes the analysis rather simple, but it makes the presentation of the approach clearer. This basic approach can be extended to include these situations. The preliminary sections discuss the derivation of the function relating the run lengths of adjacent runs for simple objects. Subsequent sections discuss the construction of a data structure, called a *knowledge frame*, which will be used to predict the orientation of a particular object.

2 DERIVATIONS FOR SIMPLE OBJECTS

In this section the equations describing the interdependence of object orientation and run lengths of successive scan lines are derived. The equations for the run lengths for rectangular, circular, and triangular objects are obtained assuming that the scan line is at an angle θ with respect to the object. Instead of rotating the object and solving for the run lengths, it is sufficient to assume that the object is fixed in a particular coordinate frame and that the scan

lines are rotated as in Figure 1. The derivations assume run lengths to take real values.

2.1 DERIVATION FOR a RECTANGULAR OBJECT

Assume that the rectangle has sides of length a and b as shown in Figure 1. The corners are located at $(0,0)$, $(a,0)$, $(0,b)$, and (a,b) so that the equations for the boundaries are: $x=0$, $y=0$, $x=a$ and $y=b$. Assume that the scanning occurs along the line $y=mx+c$. The run length for this case may be defined as the distance between the points of intersection of the scan line and any two adjacent edges of the rectangle. Consider the corner point (a,b). The point of intersection (x_1,y_1) of the scan line with the line $y=b$ is

$$(x_1,y_1)=\left(\frac{b-c}{m},b\right).$$

Similarly the point of intersection (x_2,y_2) of the scan line with the line $x=a$ is

$$(x_2,y_2)=(a,ma+c).$$

Hence the run length r_1 corresponding to this case is given by

$$\begin{aligned} r_1 &= \sqrt{((am-b+c)/m)^2+(am-b+c)^2} \\ &= (am-b+c)\sqrt{(1+m^2)/m^2}. \end{aligned}$$

For the next scan line, the slope is the same but the y-axis intercept c is different. Let this be $c-\Delta c$. For this case the run length r_2 is given by

$$r_2=(am-b+c-\Delta c)\sqrt{(1+m^2)/m^2}.$$

The difference between the two run lengths $\Delta r=r_1-r_2$ is (assuming $m=\tan\theta$)

$$\begin{aligned} \Delta r &= \Delta c\sqrt{(1+m^2)/m^2} \\ &= \Delta c\frac{\sec\theta}{\tan\theta}=\frac{\Delta c}{\sin\theta}=\frac{\Delta d}{\cos\theta\sin\theta}. \end{aligned}$$

where Δd is the normal distance between adjacent scan lines. Hence the angle of orientation θ for this rectangle is given by

$$\theta=\sin^{-1}\frac{\Delta c}{\Delta r}=\frac{1}{2}\sin^{-1}\left(\frac{2\Delta d}{\Delta r}\right). \quad (1)$$

Note that this is the result one would obtain if (a,b) is the first corner scanned. Also it can be seen that Δr is a negative quantity up to the shaded region where it becomes zero as in Figure 1, and hence θ would be negative, which verifies the result.

Let $R=\Delta r$ and $C=\Delta c$. Then Equation 1 can be rewritten as

$$\sin\theta=\frac{C}{R}.$$

Differentiating the above equation with respect to R, one obtains

$$\cos\theta\frac{\partial\theta}{\partial R}=\frac{-C}{R^2}.$$

Since $\cos\theta=\sqrt{(R^2-C^2)}/R$, it follows that

$$\frac{\partial\theta}{\partial R}=\frac{-C}{R\sqrt{R^2-C^2}}.$$

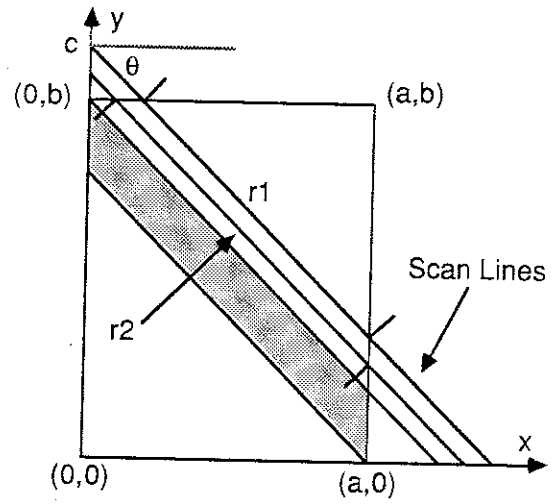

Figure 1 - Derivation for a Rectangular Object

Note that this equation is not valid when $R=C$ or when $R=0$. When $R=C$, $\theta=\pi/2$, and the scan line is parallel to the y-axis. Note that the above derivation is valid only for $0<\theta<\pi/2$. For $\theta>\pi/2$ the point $(0,b)$ is the first vertex scanned (assuming scanning to progress from the positive to negative y-axis) and the final result still holds. In fact, due to the fact that the angle subtended at each vertex is $\pi/2$, the equation for Δr holds for any orientation θ in the region between the first vertex scanned and the next vertex scanned. Note that after the scan of the second vertex the value of Δr is zero for the shaded region in Figure 1. Hence the equation for Δr can be written more generally as

$$\Delta r=\pm\Delta c\sqrt{\frac{1+m^2}{m^2}} \quad (2)$$

for the unshaded region and $\Delta r=0$ for the shaded region. When the object is a square, $b=a$ and the general expression holds for the entire region in the object.

2.2 DERIVATION FOR a TRIANGULAR OBJECT

Assume a triangle with edge equations $y=m_1x+c_1$, $y=m_2x+c_2$, and $y=m_3x+c_3$. For the sake of simplicity the coordinate axes are so chosen that one of the bases of the triangle lies along the x-axis as in Figure 2. Let (x_1,y_1) and (x_2,y_2) be the two points of intersection of the scan line with equation $y=lx+h$ and the two edges, 1 and 2, of the object. Then by definition, the run length is given by the distance between these two points. It can be seen that the points of intersection are

$$(x_1,y_1)=\left(\frac{c_2-h}{l-m_2},\frac{lc_2-hm_2}{l-m_2}\right)$$

$$(x_2,y_2)=\left(\frac{c_1-h}{l-m_1},\frac{lc_1-hm_1}{l-m_1}\right).$$

Hence it can be seen that

$$x_1-x_2=\frac{l(c_2-c_1)+h(m_1-m_2)+m_2c_1-m_1c_2}{(l-m_1)(l-m_2)}.$$

Since the points (x_1,y_1) and (x_2,y_2) lie in the same scan line with slope l, it can be seen that $y_1-y_2=l(x_1-x_2)$. Then the run length is given by

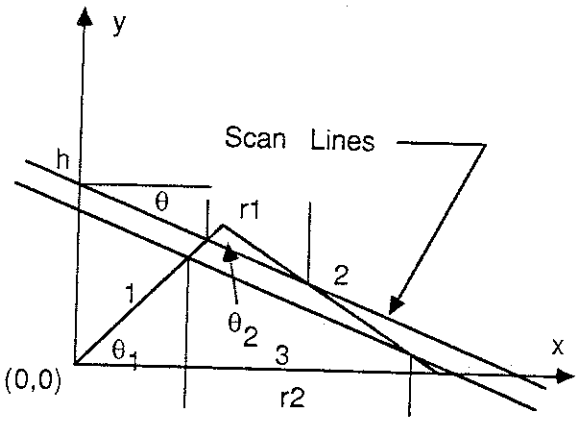

Figure 2 - Derivation for a Triangular Object

$$r_1 = (x_1 - x_2)\sqrt{1+l^2}\,.$$

Similarly r_2, the run length for the next scan line, can be obtained by substituting $h - \Delta h$ for h in the equation for $x_1 - x_2$. Hence the change in run length, Δr is given by

$$\Delta r = \frac{\Delta h(m_1 - m_2)}{(l - m_1)(l - m_2)}\sqrt{1+l^2}\,. \tag{3}$$

For the case with $l = 0$ we have

$$\frac{\Delta r}{\Delta h} = \frac{m_1 - m_2}{m_1 m_2}\,.$$

If one lets $l = \tan\theta$ and $m_1 = \tan\theta_1$, then $m_2 = \tan(\theta_2 + \theta_1)$. When $\theta_2 = \pi/2 = \theta_1$ one has

$$\frac{\Delta r}{\Delta h} = \frac{1}{\sin\theta},$$

which is the same result as in the rectangular case.

Next consider the actual measurement process, in which the scan lines are horizontal. Let (r, c) be a row-column coordinate pair. Let (r_{si}, c_{si}) and (r_{ei}, c_{ei}) denote the intersection points of the ith scan line with the object. Then the slopes M_1 and M_2 of the edge segments 1 and 2 forming the vertex with angle θ_2 in the scan coordinate system are given by the estimates

$$M_1 = \frac{1}{c_{s2} - c_{s1}}$$

$$M_2 = \frac{1}{c_{e2} - c_{e1}}\,.$$

However, M_1 and M_2 are related to slopes m_1 and m_2 in the object coordinate system by the following equations:

$$M_1 = \frac{m_1 - l}{1 + m_1 l}$$

$$M_2 = \frac{m_2 - l}{1 + m_2 l}\,.$$

Letting $\Delta c_e = c_{e2} - c_{e1}$ and $\Delta c_s = c_{s2} - c_{s1}$ and using these equations, m_1 and m_2 can be shown to be

$$m_1 = \frac{(l + 1/\Delta c_e)}{(1 - l/\Delta c_e)}$$

$$m_2 = \frac{(l + 1/\Delta c_s)}{(1 - l/\Delta c_s)}\,.$$

Substituting for m_1 and m_2 in Equation 3 and simplifying one obtains

$$\Delta r = \frac{\Delta c_s - \Delta c_e}{l^2 + 1}\,.$$

Since $l = \tan\theta$, one can see that

$$\cos\theta = \sqrt{\frac{\Delta r}{\Delta c_s - \Delta c_e}}$$

and hence that

$$\theta = \cos^{-1}\left(\sqrt{\frac{\Delta r}{\Delta c_s - \Delta c_e}}\right) \tag{4}$$

so that the orientation of the object can be predicted easily from run lengths. The next section discusses the rate of change in run length with respect to the slope of the scan line.

2.3 RATE of CHANGE of RUN LENGTH

The equation for run length can be rewritten as

$$\begin{aligned} r &= (x_2 - x_1)\sqrt{1+l^2} \\ &= XL, \end{aligned}$$

where $X = x_2 - x_1$ and $L = \sqrt{1+l^2}$. Now $\partial X/\partial l$ and $\partial L/\partial l$ are given by

$$\frac{\partial X}{\partial l} = \frac{x_1}{l - m_1} - \frac{x_2}{l - m_2}$$

$$\frac{\partial L}{\partial l} = \frac{l}{\sqrt{1+l^2}}$$

and $\partial r/\partial l$ is given by

$$\begin{aligned} \frac{\partial r}{\partial l} &= \frac{\partial X}{\partial l}L + \frac{\partial L}{\partial l}X \\ &= \frac{x_1(1 + m_1 l)}{(l - m_1)\sqrt{1+l^2}} - \frac{x_2(1 + m_2 l)}{(l - m_2)\sqrt{1+l^2}} \\ &= \left(\frac{x_1}{\tan(\theta - \theta_1)} - \frac{x_2}{\tan(\theta - \theta_2)}\right)\cos\theta\,. \end{aligned}$$

3 DERIVATION FOR ARBITRARY POLYGONS

The above discussion shows that for simple objects it is easy to obtain the relationship between run lengths of adjacent runs. However, it is difficult to obtain formally the functional form of the relationship for an arbitrary polygon. The problem may be solved by constructing a data structure which tabulates these. As an object is scanned, the scan lines cross one object vertex after another. The number of scan lines between any two vertices of an object is dependent upon the orientation of the object. The procedure attempts to identify the vertices as they are scanned. After a vertex is identified, the number of scan lines to the next vertex is predicted. Since this value varies with orientation, a table giving the number of scan lines between vertices of an object can be constructed and used for predicting the orientation.

3.1 CALCULATION of NUMBER of SCAN LINES BETWEEN ANY TWO VERTICES

The number of scan lines between any two vertices of an object can be derived using the equations of the polygon boundaries. It can be shown that the number of scan lines between the first two vertices of the triangular object shown in Figure 3 is given by

$$N_s = s_1 \sin(\theta_1 - \theta) .$$

For a triangular object the total number of scan lines intersecting the object is given by

$$N = s_1 \sin(\theta_1 - \theta) + s_3 \sin(\theta) .$$

where s_1 and s_3 are the lengths of the sides of the triangle shown in Figure 3.

Let $\{v_1, v_2, \ldots, v_{n-1}, v_n\}$ denote the set of vertices of a polygon in the object frame, and let $\{\theta_1, \theta_2, \ldots, \theta_{n-1}, \theta_n\}$ denote the set of internal angles subtended at corresponding vertices of the vertex set. Let $(x_1, y_1), \ldots, (x_n, y_n)$ denote the coordinates of vertices $v_1, \ldots, v_n$, respectively. As before, let the orientation of the scan lines be at an angle θ with respect to the horizontal axis of the object system.

The equation of the scan line passing through any point (x_a, y_a) is given by

$$y = mx + c$$

where $c = y_a - mx_a$ and $m = \tan\theta$. The y-intercept c varies with (x_a, y_a). It is therefore simple to obtain a sequence of y-intercepts due to scan lines passing through vertices $v_1, \ldots, v_n$. Let this sequence of y-intercepts be denoted $c_1, \ldots, c_n$. Note that as shown in Figure 3,

$$c_i = y_i - mx_i .$$

The number of scan lines between any two vertices (v_i) and (v_j) is given by

$$d_{ij}(\theta) = |(c_i - c_j)\cos\theta| .$$

Note that d_{ij} gives a measure of the number of scan lines between vertex j and vertex i.

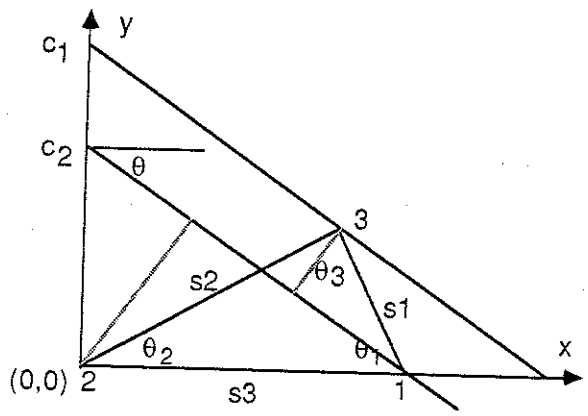

Figure 3 - Number of Scan Lines Between Vertices

3.2 ORDER of SCANNING of VERTICES

Since the object might have any orientation, the order in which the vertices are scanned is dependent on θ. This order can be obtained from the sequence of y-intercepts $c_1, \ldots, c_n$ in the object coordinate system. Let the intercepts c_i be ordered by decreasing y-values to correspond with the order in which the vertices are encountered. The scanning distance between any two vertices will then be proportional to the differences between their corresponding intercept values.

The above discussion points out that a simple procedure could be used to obtain the scanning order of vertices as well as the distance between scans of successive vertices. It should be noted that although there are $n!$ orderings of n vertices, the number that are physically possible depends upon the object geometry. It is assumed here that for most objects of interest, these orderings can be enumerated. For each such ordering there is a range of θ for which that ordering is valid. For example, assume that the object is a triangle with internal angles θ_1, θ_2, and θ_3. When $\theta \leq \theta_1$, the order of scanning of the vertices is (3,1,2). For $\theta_1 < \theta \leq \pi$ the order is (1,3,2) and so on as shown in Figure 3. Hence it is not necessary to store the orderings for each value of θ. When the object is highly symmetric so that angles cannot be distinguished, the number of orderings can be significantly reduced.

3.3 CALCULATION of NUMBER of RUNS ALONG a SCAN LINE

The above discussion does not consider the fact that several runs may exist along a scan line across a non-convex object and that the number of runs in succeeding scan lines may increase or decrease. Keeping track of this is the function of a connected components algorithm (Shapiro, et al., 1985). This makes the problem more complex since one must determine a way to provide the details about how the runs map from scan line to scan line and how the mapping varies with orientation θ. Given a particular vertex ordering, this mapping can be obtained by using algorithms similar to scan conversion algorithms in raster graphics.

3.4 IDENTIFICATION of VERTICES USING RUN LENGTHS

Since the run-time algorithm uses information about object vertices and their interconnections, it is necessary to obtain from the input image the vertex locations and identities. The first vertex is located at the point where a scan line first crosses the object. The following observation provides a general mechanism for detecting a vertex. Let $\Delta c_d = c_i - c_{i+1}$ be the horizontal difference between the endpoints (starting points) of adjacent runs in the scan coordinate system:

> **Observation** If c_d goes from a positive (negative) value to a negative (positive) value as the scanning progresses or goes from a positive (negative) value to zero, then the column corresponding to a c_d value close to zero is a vertex. (Note that this observation holds only when one is dealing with a noisefree image).

The angle subtended at a vertex can be calculated by using the equations given for the case of triangular object. This is given by the angle between the edges forming the vertex and can be shown to be

$$\theta_s = \tan^{-1}\left(\frac{M_2 - M_1}{1 + M_1 M_2}\right) .$$

where M_1 and M_2 are the slopes of those edges in the scan coordinate system. It is possible that the first scan line coincides with an edge. This means that there are at most n possible orientations for this case, and the problem is simplified.

3.5 STRUCTURE of the KNOWLEDGE FRAME

A representation scheme should be used for storing the information giving the relationships between run lengths and the orientation angle. Knowledge frames, relational tables, and production rules are various representation schemes which could be used. Knowledge frames provide a representation scheme to organize different pieces of information with regard to a particular object. The structure of a knowledge frame is given below:

Knowledge Frame
Frame Name: "Orientation Frame"
Object of interest: Orientation Angle θ
Attributes:

θ_{min}:	value
θ_{max}:	value
Order of vertices:	$(v_1,\ldots,v_n)$
Distance expressions $N(\theta)$:	$(d_{12},\ldots,d_{(n-1)n})$
Maximum runs per scan line:	value
Minimum runs per scan line:	value

It can be seen that the above frame contains details about the range of values of θ for which the vertex orderings and the number of runs are valid. The knowledge frame is constructed using output obtained from procedures outlined before. Each knowledge frame consists of several attributes like θ_{min} and θ_{max} (the minimum and maximum θ for which the order of vertices given is valid) and distance values (the distances between scans of successive vertices). Often, these distance values can be represented as simple functions of θ. The exact value of θ can be estimated using measured values of d_{12} or d_{13}. Then the measured values of $d_{34},\ldots,d_{(n-1)n}$ are compared with those in the table. A run-time algorithm which uses the information in the frame to predict the orientation is given in the next section.

3.6 ALGORITHM

At run-time the first step of the procedure is to identify the first vertex. However, since numerical accuracy is limited by camera resolution, since noise may be present, and since several object angles may be close or identical to one another, one may have to hypothesize several initial vertices. Next the number of scan lines to the next vertex (vertex 2) is measured, and a value of θ is obtained using the inverse of the distance expression associated with vertex 1. Using this estimate for θ, the distances to candidates for the next vertex (vertex 3) are calculated. Then vertex 3 is located, and the number of scan lines counted from vertex 2 to vertex 3 is compared to the distance estimates. The candidate with a distance estimate closest to the measured number of scan lines is assumed to be the successor. Using the estimated internal angle of vertex 3 and the estimated successor of vertex 2, the best candidate for vertex 3 is chosen, and then the estimate for θ is obtained using the inverse of the distance expression associated with vertex 2. This θ estimate is used and the same procedure is repeated until the distance values estimated using θ match the measured distance values closely. In the event of a contradiction at a particular node, backtracking occurs and the next best possible path is selected. The best possible path is that for which the total number of scan lines measured up to the current vertex is closest to the total number of scan lines predicted.

3.6.1 Example

Consider the polygon shown in Figure 4. The polygon has vertices a, b, c, d, e with corresponding coordinates $(0,0)$, $(1,0)$, $(2,2)$, $(2,4)$, and $(0,2)$. Let θ be the angle made by edge segment ab with the horizontal scan axis. Note that the angles subtended at vertices a, b, c, d, e are 90, 116.5, 153.4, 45, and 135 degrees, respectively. The θ ranges and their corresponding vertex orderings are given below.

Range	Order of vertices
135 to 180	(d e c a b)
118 to 135	(e d a c b)
105 to 118	(e a d b c)
90 to 105	(e a b d c)
64 to 90	(d c b e a)
0 to 64	(d c e b a)

It can be seen that changes in the orderings occur when the scan line is parallel to one of the sides of the polygon. For example, when θ is 45 degrees the edge segment de is parallel to the scan line, and when θ changes from 45 to a higher value the order becomes (e d a c b) as given in the table above. To predict the exact orientation, one needs the distances between scans of successive vertices. The following table gives the values of the number of scan lines for increments of θ of 20 degrees.

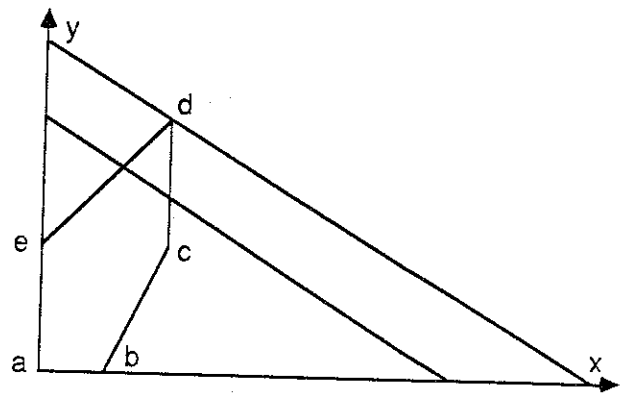

Figure 4 - Frame Example

θ	Number of scan lines
170	3.7655
150	2.9634
130	2.0515
110	1.8795
90	2.0000
70	3.2481
50	4.1037
30	4.4642
10	4.2861

The distance between any two vertices for a particular range of θ can be represented in a functional form as given below. Let (d c e b a) be the order of scanning of vertices. It can be shown in this example that the distances between vertices are given by

$$D_{dc} = 2\cos\theta$$
$$D_{ce} = -2m\cos\theta$$
$$D_{eb} = (2+m)\cos\theta$$
$$D_{ba} = -m\cos\theta\ .$$

Note that these distances were obtained by solving the following equations:

$$\begin{aligned} 4 &= 2m + C_d \\ 2 &= 2m + C_c \\ 2 &= C_e \\ -m &= C_b \\ 0 &= C_a \ . \end{aligned}$$

These equations are simply the equations of the scan line passing through the vertices a, b, c, d, e, and C_a, C_b, C_c, C_d, C_e are the corresponding y-intercepts. The distance between any two vertices i and j is given by

$$D_{ij} = |(C_i - C_j)\cos\theta| \ .$$

From these equations the knowledge frame can be constructed. It can be seen that the number of runs along a particular scan line is always one. Hence the entries for max.number of runs and the minimum number of runs will be one. The knowledge frame for this example is as given below:

Knowledge Frame
Frame name: "Orientation Frame"
Object of interest: Orientation Angle θ
Attributes:

θ_{min}:	0
θ_{max}:	45
Order of vertices:	(d c e b a)
Distance expressions:	$(2\cos\theta, -2m\cos\theta,$ $(2+m)\cos\theta, -m\cos\theta)$
Maximum runs per scan line:	1
Minimum runs per scan line:	1

It can be seen that the distance expressions are dependent on θ, and hence often the algorithm predicts the right answer after calculating the distances between the first, second and third vertices. For example, the distance value $D_{ce} = -2\sin\theta$, and an estimate for θ is $\theta = \sin^{-1}(-D_{ce}/2)$.

4 CONCLUSIONS

This paper discusses an algorithm which is used to predict the orientation of an object using the relationship between run lengths and the orientation angle θ. A knowledge frame describing the run length-θ relationship is used by the algorithm for predicting θ. It was shown that the frame could be constructed rather easily using simple procedures. Since the computation of the knowledge frame is offline, the algorithm for predicting orientation is computationally efficient. Also, when the number of vertices is small, as is often the case, the memory utilized is small. Moreover, the algorithm can usually arrive at the orientation within a few scans, and it uses subsequent scans to confirm the value of θ. Additional work will be required to formulate the frame-based reasoning process, to implement a reliable vertex detector, to work out the additional details for non-convex objects, to simulate the entire algorithm, and to interface the system to a CAD/CAM design facility.

REFERENCES

Agin, G.J., and Duda, R.O. (1975). SRI vision research for advanced industrial automation. In *Proc. 2nd USA-Japan Computer Conference* (pp. 113–117).

Foith, J.P. (1982). *Intelligente Bildsensoren zum Sichten, Handhaben, Steuern, und Regeln.* New York: Springer-Verlag.

Gleason, G.J. and Agin, G.J. (1979). A modular vision system for sensor controlled manipulation. In *Proc. 9th Intl. Symposium on Industrial Robotics*, (pp. 57–70).

Henderson, T., Hansen, C., Samal, A., Ho, C.C., and Bhanu, B. (1986). CAGD-based 3-D visual recognition. In *Proc. 8th Intl. Conference on Pattern Recognition*, (pp. 230–232).

Reinhold, A.G. and VanderBrug, G.J. (1980). Robot vision for industry - the Autovision system. *Robotics Age, 2* (3), 22–28.

Shapiro, L.G., Lee, James S., and Haralick, R.M. (1985). *A connected components computer for real time image processing* (Tech. Report). Ann Arbor, MI: Machine Vision International.