# Boundary-based corner detection using ANNs 

D. M. Tsai<br>Machine Vision Lab.<br>Department of Industrial Engineering and Management<br>Yuan-Ze University, Chung-Li, Taiwan, R.O.C.<br>E-mail: iedmtsai@saturn.yzu.edu.tw<br>(Figures are not available in this report.)

## 1. INTRODUCTION

Corner detection in digital images has been shown to be extremely useful in many computer vision applications. Since information about a shape is concentrated at the corners, they prove to be practical descriptive primitives in shape representation, image interpretation, object recognition $[1,2,3,4]$ and motion analysis $[5,6]$.

Corners on a curve arise where two relatively straight line segments intersect. Corners in a gray-level image occur in regions of rapidly changing intensity levels and gradient directions. Therefore, corner detection techniques can be classified into two categories: binary and gray level. The binary approaches involve, first segmenting the image into meaningful regions, and then extracting boundaries from the regions of interest. Corners on the extracted boundaries are typically identified for those points with high curvature. Methods of discrete curvature measurement [7, 8, 9] are based on the rate of change of tangent direction. A variety of curvature measurement methods have also been reported by Teh and Chin [10]. The boundary-based corner detection methods [11-17] generally are easy to implement and computationally fast. However, when the curve is rotated in different orientations, they may suffer from poor detection due to the instability of discrete curvature measurement in digital images.

The gray level approaches directly work on the gray level image without the need for image segmentation. They can be further divided into two groups, namely template-based and gradient-based techniques. Template-based corner detection [18, 19] involves determining the similarity between a given template of a specific angle and all subwindows of the same size as the emplate in a gray-level image. The cornerity measure at a pixel is obtained by computing the correlation or convoluting with each of the templates and selecting the maximum value. Template-based schemes are computationally expensive since many different templates must be applied at each point in the image - one for each desired corner shape in each possible orientation. Gradient-based corner detection [20, 21, 22, 23] relies on measuring the
curvature of an edge that passes through a neighborhood in a gray-level image. The measure of cornerity is expressed as a function of the edge strength and the gradient of edge direction. Gradient-based methods are more likely to response to noise and often are worse at localizing corners.

This paper deals with the first category of corner detection approaches. We develop a new boundary-based corner detection method using artificial neural network (ANN) techniques. The goal of the proposed ANN corner detection is to possess robust detection and localization regardless of object orientations in the image. Two ANN models are proposed. The first network model is for the detection of corner points having high curvature, and the second model is for the detection of tangent points and inflection points that generally have low curvature.

Any boundary point within a small neighborhood window on the curve can be considered as the intersection point of two relatively straight line segments. Therefore, for a given boundary point $p_{i}$, the angles of the forward arm (i.e., a set of neighboring points posterior to $p_{i}$ ) and the backward arm (i.e., a set of neighboring points prior to $p_{i}$ ) with respect to the x -axis are evaluated using the first ANN. The curvature at the point is associated with the angle between the forward arm and backward arm. The point with local maximum curvature is identified as a corner.

A tangent point where a line is tangent to a circle and an inflection point connecting a convex arc and a concave arc are typical feature points of most manmade objects. These points of transition must be extracted in a reliable way to split shapes into meaningful segments for shape analysis. However, most of existing corner detection methods can not identify these two feature points due to their small curvature values. In this study, we develop a second ANN that recognizes the points of tangency and inflection by the pattern of curvature signs over a small neighborhood window rather than the curvature magnitude. With the use of both ANN models, all features of corners, tangent points and inflection points can be extracted from the boundary of any arbitrary shape. The proposed ANN models provide a great degree of robustness and fault tolerance. They reduce the effect of quantization and curves in different orientations, and minimize the number of spurious corners to make the corner detection stable and effective in computer vision tasks.

This paper is organized as follows: Section 2 introduces the first ANN model for detecting corners with high curvature. Section 3 discusses the second ANN model for detecting tangent points and inflection points. Section 4 presents the
experimental results. The conclusion is reached in section 5 .

## 2. ANN FOR DETECTING CORNERS

ANNs are specified by the topology of the network, the characteristics of the nodes and the processing algorithm. In this study, we use back-propagation [24] to construct the first ANN model for detecting corners with high curvature. The proposed back-propagation neural network comprises an input layer, two hidden layers and one output layer. Each layer is fully connected to the succeeding layer. The outputs of nodes in one layer are transmitted to nodes in another layer through links. The link between mdes indicates flow of information during recall. During learning, information is also propagated back through the network and used to update the connection weights between nodes.

Let $O_{j}$ be the output of the j -th node in the previous layer and $w_{i j}$ the connection weight between the $i$-th node in one layer and the $j$-th node in the previous layer. The total input to the $i$-th node of a layer is given by

$$
\text { net }_{i}=\sum_{j} w_{i j} \cdot O_{j}
$$

The output of the node $i$ is determined by a hyperbolic tangent activation function

$$
O_{i}=f\left(\text { net }_{i}\right)=\frac{e^{n e e_{i}}-e^{-n e t_{i}}}{e^{n e t_{i}}+e^{-n e t_{i}}}
$$

Let $O_{n j}$ be the evaluated output of node $j$ in the output layer for input pattern $n$, and $D_{n j}$ the desired output of node $j$ for input pattern $n$. For a given training set of $N$ input patterns, the squared error of the system can be written as

$$
E=\frac{1}{2 N} \sum_{n} \sum_{j}\left(D_{n j}-O_{n j}\right)^{2}
$$

The generalized $\delta$-rule learning algorithm [25] with training by epoch [24] is applied to adjust the weights so that the error $E$ is a minimum. A detailed derivation of the learning procedure can also be found in reference [24].

### 2.1. Learning phase

The first ANN model uses a set of angular patterns of known angles in a training database as input for supervised learning. Let the sequence of $n$ digital points describe a closed curve $P$,

$$
P=\left\{p_{i}=\left(x_{i}, y_{i}\right), i=1,2, \ldots \ldots, n\right\}
$$

where $p_{i+1}$ is a neighbor of $p_{i}$ (modulo $n$ ), and $\left(x_{i}, y_{i}\right)$ are the Cartesian
coordinates of $p_{i}$.

Any boundary point $p_{i}$ within a small region of support can be considered as the intersection of two relatively straight line segments. Define the directed forward arm of $p_{i}, F_{i}$, as a set of $u$ neighboring points (including $p_{i}$ itself) posterior to $p_{i}$, i.e.,

$$
F_{i}=\left\{p_{i}, p_{i+1}, \ldots . ., p_{i+u-1}\right\}
$$

Here, $u$ specifies the size of region of support. Similarly, the directed backward $\operatorname{arm}$ of $p_{i}, B_{i}$, is a set of $u$ neighboring points prior to $p_{i}$, i.e.,

$$
B_{i}=\left\{p_{i}, p_{i-1}, \ldots \ldots ., p_{i-u+1}\right\}
$$

For a small region of support $u$, the forward arm and backward arm of a point can be approximated by two straight line segments. Therefore, the included angle between the forward arm and backward arm indicates the significance of curvature. It can be calculated by the difference between the angle of the forward arm with respect to the x -axis and the angle of the backward arm with respect to the x -axis. In this study, the training pattern of a given angle $\theta$ with respect to the $x$-axis is generated from the coordinates of $u$ digital points on a line segment with slope angle $\theta$. Since a directed forward or backward arm can be rotated between 0 and 360 degrees, the number of training patterns to the back-propagation neural network must be sufficient to represent all possible orientations of a forward or backward arm. The procedure to generate all required angular patterns is as follows:

For $0^{\circ} \leq \theta_{\mathrm{I}}<90^{\circ}$, compute the coordinates of the line segment with a given slope angle $\theta_{I}$ (see Figure 1),

$$
\begin{gathered}
y_{i}=\left[x_{i} \cdot \tan \theta_{I}\right] \\
x_{i}=0,1,2, \ldots \ldots, u-1
\end{gathered}
$$

where $u$ is the region of support, and $[\cdot]$ represents the integer round-off operation. $\left(x_{i}, y_{i}\right)$ are taken in integer to simulate the coordinate grid of a digital image. $\left(x_{i}, y_{i}\right)$ are further transformed to the values between 0 and 1 as the standard input to the neural network. Therefore, the angular pattern of $\theta_{I}$, denoted by $A_{\theta_{I}}$, is given by

$$
A_{\theta_{I}}=\left\{\left(x_{i}^{\prime}, y_{i}^{\prime}\right) \left\lvert\, x_{i}^{\prime}=\frac{x_{i}}{u}\right., y_{i}^{\prime}=\frac{y_{i}}{u}, y_{i}=\left[x_{i} \cdot \tan \theta_{I}\right], x_{i}=0,1,2, \ldots \ldots, u-1\right\}
$$

For $90^{\circ}<\theta_{\text {II }} \leq 180^{\circ}$, the angular pattern $A_{\theta_{\|}}$is the reflection of $A_{\theta^{\prime}}$ about the $y$-axis, i.e.,

$$
A_{\theta_{I}}=\left\{\left(-x_{i}^{\prime}, y_{i}^{\prime}\right) \mid\left(x_{i}^{\prime}, y_{i}^{\prime}\right) \in A_{\theta_{I}}, \forall i\right\}
$$

For $180^{\circ}<\theta_{\text {III }} \leq 270^{\circ}$, the angular pattern $A_{\theta_{I I}}$ is the reflection of $A_{\theta_{I}}$ about the x -axis, i.e.,

$$
A_{\theta_{I I}}=\left\{\left(-x_{i}^{\prime},-y_{i}^{\prime}\right) \mid\left(x_{i}^{\prime}, y_{i}^{\prime}\right) \in A_{\theta_{I}}, \forall i\right\}
$$

Finally, for $270^{\circ}<\theta_{\text {IV }}<360^{\circ}$, the angular pattern $A_{\theta_{V V}}$ is the reflection of $A_{\theta^{\prime}}$, about the x -axis, i.e.,

$$
A_{\theta_{V N}}=\left\{\left(x_{i}^{\prime},-y_{i}^{\prime}\right) \mid\left(x_{i}^{\prime}, y_{i}^{\prime}\right) \in A_{\theta_{I}}, \forall i\right\}
$$

In this study, the region of support $u$ that determines the lengths of the forward arm and backward arm is fixed for objects with similar size features. For multiplesize features in an image, the region of support used for the computation of angle $\theta$ at each boundary point could be adaptively determined based on the local properties of the point. Corner detection for multiple-size features is not the main focus of this study. Algorithms for determining adaptive region of support at each point can be found in [10, 16, 26].

It has been observed [8] that a value of region of support greater than the length of a side of a polygon ensures that no curvature maxima are found, and a vertex of the polygon will be indistinguishable from its circumscribing circle. Too large the region of support makes the forward arm (or backward arm) of a given point not representable with a straight line segment and degrades the accuracy of angular measurement, whereas too small value of $u$ will detect many false alarms and be sensitive to quantization and noise. Therefore, the region of support $u$ used in this study is appropriately set to 9 . A line segment of length 9 pixels is sufficient to specify a slope angle with $5^{\circ}$ resolution in digital images. Thus the angular patterns used in training are between $0^{\circ}$ and $360^{\circ}$ in $5^{\circ}$ increments. A total of 72 angular patterns is used during the learning phase of the neural network.

The system architecture of the four-layer back-propagation ne ural network used for detecting corners with high curvature is illustrated in Figure 2. The input layer of the network contains 18 nodes corresponding to the x -coordinates and y -coordinates of the 9 boundary points of a forward arm (or a backward arm). Two hidden layers are used in the network. The first hidden layer has 9 nodes and the second hidden layer has 4 nodes. The output layer contains a single node, which outputs the angle $\theta$ responding to a specific angular pattern $A_{\theta}$, and $0^{\circ} \leq \theta<360^{\circ}$.

Let $P=\left\{p_{i}=\left(x_{i}, y_{i}\right), i=1,2, \ldots \ldots . n\right\}$ be the boundary of an object in the input image. $\quad F_{i}$ and $B_{i}$ are the forward arm and backward arm, respectively, of a boundary point $p_{i}$ as defined previously. Note that the coordinates of all angular patterns used in the learning phase of the network start at the origin $(0,0)$, and are transformed in the range $[0,1]$. Therefore, during the recognition phase of the network, the coordinates of the forward arm $F_{i}$ and backward arm $B_{i}$ of a given boundary point $p_{i}$ are transformed as follows:

$$
\left[\begin{array}{l}
x_{j}^{\prime} \\
y_{j}^{\prime}
\end{array}\right]=\frac{1}{u}\left(\left[\begin{array}{l}
x_{j} \\
y_{j}
\end{array}\right]-\left[\begin{array}{l}
x_{i} \\
y_{i}
\end{array}\right]\right)
$$

where $j=i, i+1, \ldots \ldots, i+u-1$ for the forward arm of $p_{i}$;
$j=i, i+1, \ldots \ldots, i-u+1$ for the backward arm of $p_{i}$.

For a region of support $u=9$, the input vector of a boundary point $p_{i}$ to the network is given by

$$
\left(x_{i}^{\prime}, y_{i}^{\prime}, x_{i+1}^{\prime}, y_{i+1}^{\prime}, \ldots . ., x_{i+8}^{\prime}, y_{i+8}^{\prime}\right) \text { for the forward arm, }
$$

and

$$
\left(x_{i}^{\prime}, y_{i}^{\prime}, x_{i-1}^{\prime}, y_{i-1}^{\prime}, \ldots \ldots, x_{i-8}^{\prime}, y_{i-8}^{\prime}\right) \text { for the backward arm. }
$$

Figure 1. Generating the angular pattern from a line segment with slope angle $\theta$

Figure 2. The system architecture of the first ANN for detecting corners.
(Only partial connection links are illustrated.)

Let $\theta_{f_{i}}$ and $\theta_{b_{i}}$ be the output values (the angles with respect to the x -axis) of the ANN responding to the input vectors of the forward arm $F_{i}$ and backward arm $B_{i}$ of a boundary point $p_{i}$, respectively. The included angle $\psi_{i}$ between the forward arm and backward arm of point $p_{i}$ is
$\psi_{i}= \begin{cases}\theta_{f_{i}}-\theta_{b_{i}} & \text { if }\left(\theta_{f_{i}}>\theta_{b_{i}}\right) \text { and }\left(\theta_{f_{i}}-\theta_{b_{i}}<180^{\circ}\right) \\ 360^{\circ}-\left(\theta_{f_{i}}-\theta_{b_{i}}\right) & \text { if }\left(\theta_{f_{i}}>\theta_{b_{i}}\right) \text { and }\left(\theta_{f_{i}}-\theta_{b_{i}}>180^{\circ}\right) \\ \theta_{b_{i}}-\theta_{f_{i}} & \text { if }\left(\theta_{f_{i}}<\theta_{b_{i}}\right) \text { and }\left(\theta_{b_{i}}-\theta_{f_{i}}<180^{\circ}\right) \\ 360^{\circ}-\left(\theta_{b_{i}}-\theta_{f_{i}}\right) & \text { if }\left(\theta_{f_{i}}<\theta_{b_{i}}\right) \text { and }\left(\theta_{b_{i}}-\theta_{f_{i}}>180^{\circ}\right)\end{cases}$
and $0^{\circ} \leq \psi_{i} \leq 180^{\circ}$.

The included angle $\Psi_{i}$ indicates the significance of a corner point. The smaller the included angle, the stronger the evidence that the boundary point is a corner. In order to follow the conventional definition of curvature, i.e., the larger the value of the measure, the higher the curvature, the curvature used in this study is defined to be the supplementary angle of the included angle. Hence, the curvature at a point $p_{i}, \kappa_{i}$, is given by

$$
\kappa_{i}=180^{\circ}-\psi_{i}, \quad 0^{\circ} \leq \kappa_{i} \leq 180^{\circ}
$$

An acute corner has small included angle and high curvature, whereas an obtuse corner has large included angle and low curvature. Curvature $\kappa$ gives the measure of relative significance at each boundary point. A point $p_{i}$ is said to be a corner if its curvature $\kappa_{i}$ exceeds a predetermined threshold, and individual corners are separated by a spacing of at least $u$ points. Figures 3(a) and 3(b) present the binary images of a $30^{\circ}-60^{\circ}-90^{\circ}$ triangle in two different orientations. Figures 3(c) and 3(d) show the corresponding included angles along the boundaries. Note that the plots shown in Figures 3(c) and 3(d) resemble to each other regardless of the orientations of the triangle. The proposed ANN generates approximately $60^{\circ}, 90^{\circ}$ and $30^{\circ}$ for the three corners $\mathrm{A}, \mathrm{B}$ and C of the triangle as shown in Figure 3. The remaining noncorner points on the boundary of the triangle have included angles approximate to $180^{\circ}$.

## 3. ANN FOR POINTS OF TANGENCY AND INFLECTION

Curvilinear curves containing piecewise linear segments and circular arcs dominate the shapes of most man-made objects. It is not uncommon to find the points of tangency and inflection on a curvilinear curve. A tangent point is a point of transition between an arc and a line tangent to the arc. An inflection point is a point of transition between a convex arc and a concave arc. These two feature points make discontinuous change in direction, and are as important as corner points with high curvature for shape analysis. However, the proposed ANN aforementioned and any existing corner detection methods using measures of curvature can not reliably identify tangent points and inflection points due to their relatively low curvature. Figure 4 shows a curvilinear curve with 6 linear and circular segments. Corner points $1,2,3$ and 4 can be easily identified using the ANN proposed previously. However, tangent point $P$ and inflection point $Q$ must be identified using the method other than measures of curvature.

In real Euclidean space, we observe that the curvature of a straight line is always zero. The curvature of a circular arc is a non-zero constant, and the curvatures of a convex arc and a concave arc have opposite signs. Therefore, the sign pattern of curvatures within a small neighborhood window of a boundary point is a distinguishable representation for tangent points and inflection points. The sign pattern of a tangent point is a step-shape curve. The sign pattern of an inflection point is also a step-shape curve but with a zero crossing of curvature. The sign patterns of other boundary points are a line or a line with a spike. Assume that a convex arc has positive curvature, and a concave arc has negative curvature. Figures 5(a) and 5(b) show the sign patterns of a tangent point and an inflection point, respectively.

Figure 3. A $30^{\circ}-60^{\circ}-90^{\circ}$ triangle and its included angles along the boundary in two different orientations.

Figure 4. A curvilinear object with corners, tangent point and inflection point.

These two sign patterns are distinctive from the sign patterns of other boundary points such as the intersection point of two line segments (Figure 5(c)) and the point on a circular arc (Figure 5(d)). In this study, a " +1 " (positive one) is assigned to the boundary point on the convex arc, a " 1 " (negative one) is assigned to the boundary point on the concave arc, and a " 0 " (zero) is assigned to the point on a straight line. The sign of curvature at a point is determined as follows:

Let $\theta_{f_{i}}$ and $\theta_{b_{i}}$ be the angles of the forward arm and backward arm of a boundary point $p_{i}$, which are given by the outputs of the first ANN aforementioned. $\theta_{f_{i}}$ and $\theta_{b_{i}}$ are transformed between $0^{\circ}$ and $180^{\circ}$. Hence

$$
\begin{aligned}
& \theta_{f_{i}}^{\prime}= \begin{cases}\theta_{f_{i}} & \text { if } \theta_{f_{i}} \leq 180^{\circ} \\
\theta_{f_{i}}-180^{\circ} & \text { if } \theta_{f_{i}}>180^{\circ}\end{cases} \\
& \theta_{b_{i}}^{\prime}= \begin{cases}\theta_{b_{i}} & \theta_{b_{i}} \leq 180^{\circ} \\
\theta_{b_{i}}-180^{\circ} & \text { if } \theta_{b_{i}}>180^{\circ}\end{cases}
\end{aligned}
$$

Define

$$
\kappa_{i}^{\prime}=\theta_{f_{i}}^{\prime}-\theta_{b_{i}}^{\prime}
$$

Let $\zeta_{i}$ be the sign of curvature at the boundary point $p_{i}$. Then,

$$
\varsigma_{i}=\left\{\begin{array}{cc}
0 & \text { if }\left|\kappa_{i}^{\prime}\right| \leq T_{s} \\
+1 & \text { if } \kappa_{i}^{\prime}>T_{s} \\
-1 & \text { if } \kappa_{i}^{\prime}<-T_{s}
\end{array}\right.
$$

To accommodate quantization error, the threshold $T_{s}$ is used to determine the sign of curvature. In this study, the threshold $T_{s}$ is set to $10^{\circ}$ since the angular
patterns used in training for the first ANN are in $5^{\circ}$ increments.
(a)
(b)
(c)
(d)

Figure 5. Sign patterns of curvatures for (a) a tangent point, (b) an inflection point, (c) an intersection point of two line segments, and (d) a point on the circular arc.

For a given region of support $u$, the sign pattern of curvatures of a boundary point $p_{i}$ in its vector form is given by

$$
\left(\varsigma_{f_{i-l}}, \varsigma_{f_{i-u+1}}, \ldots \ldots, \varsigma_{f_{i-1}}, \varsigma_{b_{i+1}}, \varsigma_{b_{i+2}}, \ldots . ., \varsigma_{b_{i+u-1}}, \varsigma_{b_{i+1}}\right)
$$

where $\zeta_{f_{i-j}}$ and $\zeta_{b_{i+j}}$ are the curvature signs of the forward arm and backward arm at point $p_{i}$, respectively, and $\zeta_{f_{i-j}}$ and $\zeta_{b_{i+j}} \in\{-1,0,1\}, j=1,2, \ldots \ldots$., $u$.

In this study, the size of region of support is taken to be 9. Therefore, the sign patterns of tangent points and inflection points are given below.

Sign patterns for tangent points:
1). ( $1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0)$
2). $(-1,-1,-1,-1,-1,-1,-1,-1,-1,0,0,0,0,0,0,0,0,0)$
3). $(0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1)$
4). $(0,0,0,0,0,0,0,0,0,-1,-1,-1,-1,-1,-1,-1,-1,-1)$

Sign patterns for inflection points:
1). $(-1,-1,-1,-1,-1,-1,-1,-1,-1,1,1,1,1,1,1,1,1,1)$
2). $(1,1,1,1,1,1,1,1,1,-1,-1,-1,-1,-1,-1,-1,-1,-1)$

The sign patterns derived above are used as the training samples to the second back-propagation neural network for detecting tangent points and inflection points. The topology of the second ANN consists of one input layer with 18 input nodes (corresponding to 9 points of the forward arm and 9 points of the backward arm), one single hidden layer with 9 nodes, and one output layer with two output nodes. The first output node responds to tangent points and the second output node responds to inflection points. The output responses are in the range of $[0,1]$. The system architecture of the second ANN is shown in Figure 6.

Since the input value to the second ANN is the binary sign of curvature at a point rather than the analog magnitude of curvature, the representations of tangent points

Figure 6. The system architecture of the second ANN for detecting tangent points and inflection points. (Only partial connection links are presented. )
and inflection points are robust and affected less by quantization and rotation of objects. The output of the first four-layer back-propagation ANN is applied to compute the curvature at a boundary point. It is also applied to the second threelayer back-propagation ANN to evaluate the sign of curvature at each point. With the use of both proposed ANNs, all feature points of corners, tangency and inflection on arbitrary curves can be identified.

## 4. EXPERIMENTAL RESULTS

In this section, we present experimental results to evaluate the performance of the ANN corner detectors. The robustness of the proposed methods is addressed by the following performance criteria:
1). Detection: The number of true corners should be detected as many as possible while the number of spurious corners is minimized.
2). Localization: The corners should be detected as close as possible to their true locations. It is measured by the distance (in pixels) between the detected corner point and the true corner specified by a human viewer.

Let $C=\left\{c_{1}, c_{2}, \ldots \ldots ., c_{m}\right\}$ be a set of $m$ desired corner points extracted from a boundary $P=\left\{p_{1}, p_{2}, \ldots \ldots, p_{n}\right\}$. Denote $N\left(c_{i}\right)$ by a set of boundary points in the neighborhood (defined by the region of support $u$ ) of corner point $c_{i}$, The detection capability is measured by the curvature ratio $\Re$, which is defined as

$$
\mathfrak{R}=\frac{\operatorname{Min}\left\{\kappa\left(c_{i}\right) \mid c_{i} \in C, \forall i\right\}}{\operatorname{Max}\left\{\kappa\left(p_{j}\right) \mid p_{j} \in P, p_{j} \notin N\left(c_{i}\right), \forall i, j\right\}}
$$

where $\kappa\left(c_{i}\right)$ is the curvature at the points $c_{i} . \quad p_{j}$ represents any boundary point not in the neighborhood of the corner point $c_{i}$, and $\kappa\left(p_{j}\right)$ is the curvature at the point $p_{j}$. The numerator of $\mathbb{K}$ is the minimum curvature value among all desired corners, and the denominator of $\mathfrak{R}$ is the maximum curvature value of noncorner points on the boundary. The measure of $\mathfrak{R}$ represents the detectability of corners and can also be indicated as the sensitivity of threshold setting. If $\Re \leq 1$, it indicates that some desired corners will be missed for a tight threshold, or spurious corners will be detected for a loose threshold. Clearly, the larger the value of $\mathfrak{R}$, the less sensitive of the threshold setting for true corner detection.

In our implementations, all algorithms are programmed in C and executed on a personal computer with a $486-\mathrm{DX} 250 \mathrm{MHz}$ processor. The image size is $480 \times 440$ in pixels. Based on the study of Liu and Srinath [14], the Freeman-Davis-based algorithm [11, 12] has outperformed other boundary-based corner detection schemes. Therefore, the performance of the proposed ANN corner detectors is compared with that of the Freeman-Davis method [11]. In the Freeman-Davis scheme, a corner is defined as an isolated discontinuity (local curvature) in the mean slope, its prominence being proportional to the length of the discontinuity-free regions to either side of the point as well as the measured magnitude of the discontinuity.

The experiments are divided into two parts: one for detecting corners with high curvature, and the other for detecting tangent points and inflection points. They are discussed individually in the following subsections.

### 4.1. Experiments on corners

Three artificial objects (Figures 7, 8 and 9) as well as a real image(Figure 13) have been used to test the performance of the first ANN for detecting corners with high curvature. Object 1 shown in Figure 7 is a polygon with vertices of angles ranging from $10^{\circ}$ to $170^{\circ}$ in $10^{\circ}$ increments. It is chosen because its corner points are well defined and can be used to evaluate the ANN's detectability for corners with various angles. The polygonal object has been tested under three different scales 75\%,
(a)Scaling factor $=75 \%$
(b)Scaling factor $=100 \%$
(c)Scaling factor $=125 \%$
(Number of boundary
points $=647$ )
(Number of boundary
points $=857$ )
(Number of boundary
points $=1059$ )

Figure 7. Test object 1 with three scales. The numbers shown in (c) represent the angles of vertices.
(a)Scaling factor $=75 \%$
(Number of boundary
points $=758$ )
(b)Scaling factor $=100 \%$
(Number of boundary
points $=1013$ )
(c)Scaling factor $=125 \%$
(Number of boundary points $=1254$ )

Figure 8. Test object 2 with three scales.
(a)Scaling factor $=75 \%$
(b)Scaling factor $=100 \%$
(c)Scaling factor $=125 \%$
(Number of boundary
(Number of boundary
(Number of boundary
points $=714$ )
points $=952$ )

Figure 9. Test object 3 with three scales.
$100 \%$ and $125 \%$ in terms of area change, and the object of each scale is rotated in eight directions at $45^{\circ}$ intervals so that the effects of scale and orientation on detected corners can be analyzed. Tables 1 (a), 1(b) and 1 (c) show the curvature rations $\mathbb{R}$ of the Freeman-Davis method and the proposed ANN detector in various orientations for scaling factors $75 \%, 100 \%$ and $125 \%$, respectively. The results shown in the tables are based on the desired corners specified by a human viewer as shown in Figure 10(a), where the corners of angles up to $160^{\circ}$ are selected. Both FreemanDavis and ANN methods fail to detect the subtle corner of angle $170^{\circ}$. However, the proposed ANN detector can reliably detect all corners up to $160^{\circ}$ for the polygonal object. The overall averages of curvature ratios for the Freeman-Davis method and the ANN detector are 1.23 and 1.38, respectively. In terms of the standard deviation of $\Re$ in different orientations, the ANN detector has consistently shown its stability for orientation changes under three different scales. Of the 24 experiments ( 3 scales, each with 8 directions), the Freeman-Davis method has 8 curvature ratios less than 1, whereas the proposed ANN has only one $\mathfrak{R}$-value smaller than 1 . A $\mathfrak{R}$-value less than 1 means spurious corners will be detected in order to extract all desired corners (as seen in figures $11(\mathrm{~b})$ and $12(\mathrm{~b})$ ). Therefore, the ANN detector is more reliable than the Freeman-Davis method for the object with orientation and scale changes.

Figures 8 and 9 show additional test objects 2 and 3 . These two curved-shape objects are taken from [14]. Each object is also experimented under three scaling factors $75 \%, 100 \%$ and $125 \%$. The objects of each scale are tested in eight different orientations varying from $0^{\circ}$ to $315^{\circ}$ at $45^{\circ}$ intervals. Tables 2 and 3 present the average curvature ratios of each object for the Freeman-Davis method and the proposed ANN method. The results are based on the desired corners specified by a human viewer as shown in Figures 11(a) and 12(a).

Since objects 2 and 3 involve arc segments with different radii, the corner detection methods are more sensitive to orientation and scale changes. With the Freeman-Davis method, all average curvature ratios $\bar{\Re}$ for both objects 2 and 3 are very small, between 0.06 and 0.21 , and none of the 24 experiments for each object result in $\mathfrak{R}$-values larger than 1 . However, with the proposed ANN method, all average curvature ratios for both objects 2 and 3 are larger than 1 . Of the 24 experiments for each object, object 2 has $17 \mathfrak{R}$-values larger than 1 , and object 3 has $22 \Re$-values larger than 1 .

In terms of localization, Table 4 summarizes the average distance (in pixels) between a detected corner point and its corresponding true corner for objects 1,2 and

3 under three scaling factors $75 \%, 100 \%$ and $125 \%$ and four orientations $45^{\circ}, 135^{\circ}$, $225^{\circ}$ and $315^{\circ}$. Table 4 reveals that the proposed ANN has generated accurate locations of corners regardless of orientation and scale changes, where the overall average distance is only 0.78 pixels, and the standard deviation of distances is less than 0.3 pixels.

Figures 13(a)-(c) show the silhouettes of a real leaf under scaling factors $75 \%$, $100 \%$ and $125 \%$. The detection results for the leaf image at the scale of $100 \%$ are presented in Figure 14. With the Freeman-Davis method, the average curvature ratio $\mathfrak{K}$ is only 0.29 and none of the 24 experiments for the real leaf images result in $\mathfrak{K}_{-}$ values larger than 1 . However, with the proposed ANN method, the average curvature ratio $\mathfrak{R}$ is 1.05 , and 16 of the 24 experiments have $\mathfrak{R}$-values larger than 1 . The proposed ANN has also generated good localization of 0.83 pixels on average, compared to 1.60 pixels of the Freeman-Davis method.

The experiments above have shown that the first ANN compares favorably with the Freeman-Davis method. It has generated good detection and localization for polygonal and curved objects in arbitrary orientation under moderate scale changes.

### 4.2. Experiments on tangent and inflection points

Three objects with tangent points and inflection points on the boundaries are used to evaluate the performance of the second ANN. These three objects, denoted as objects 4,5 and 6, are shown in Figures 15, 16 and 17, respectively. Object 6 is a real aluminum casting part. The backlighting technique is used to obtain the silhouette of the casting part, as shown in Figure 17(b). Each object is evaluated under three scaling factors $75 \%, 100 \%$ and $125 \%$, and four orientations $45^{\circ}, 135^{\circ}$, $225^{\circ}$ and $315^{\circ}$. Both the Freeman-Davis method and the first ANN detector have failed to identify the tangent points and inflection points on the boundaries of objects 4, 5 and 6. However, the second ANN has reliably detected one tangent point and two inflection points for object 4 (Figure 15(b)), and four tangent points for object 5 (Figure 16(b)) regardless of the orientation and scale changes. The tangent points of object 6 (Figure 17(b)) are less obvious since the shape of object 6 is the profile of a real aluminum casting part. The second ANN detector has shown its robustness to detect three tangent points on the boundary of object 6, as seen in Figure 17(c), even though the boundary is composed of both circular arcs and non-circular curve segments. The output responses of the second ANN for tangent points and inflection points are close to 1 , whereas all other non-tangent and non-inflection points have very
low responses close to zero.

In terms of localization, the experiments on objects 4,5 and 6 have shown that the average location errors for a tangent point and an inflection point are 2 pixels and 6 pixels, respectively. The second ANN detector has good localization for tangent points. Orientation and scale changes of an object shape have more significant impact on the locations of inflection points. This is due to the fact that a small segment within the neighborhood of an inflection point is approximate to a line and, therefore, the true transition point between two arcs is not explicitly specified.

## 5. CONCLUSION

In this paper, we have presented a new corner detection approach using artificial neural network techniques. Two ANNs are developed, one for detecting corners with high curvature, and the other for detecting points of tangency and inflection. The first ANN uses the normalized coordinates of the forward arm or backward arm of a boundary point as the input vector. The output feature of the first ANN is the angle of the forward arm (or backward arm) with respect to the $x$-axis. The curvature at a point is defined as the supplementary angle of the included angle between the forward arm and backward arm. Experimental results have revealed that the first ANN detector has good detection and localization for polygonal and curved objects in arbitrary orientations and with moderate scale changes.

Since the feature points of tangency and inflection on a boundary have relatively low curvature, the sign pattern of curvature instead of the magnitude of curvature is used as the input vector to the second ANN detector. The curvature sign at each boundary point is determined based on the outputs (angles of the forward and backward arms) of the first ANN. Both sign patterns of a tangent point and an inflection point are step-shape curves, whereas the sign patterns of other boundary points are approximately a line or a line with a spike. Experimental results have shown that the second ANN has good detection capability. The average location errors for tangent points and inflection points are 2 pixels and 6 pixels, respectively. In most cases, the proposed ANNs tend to perform corner detection close to human performance.

In this study, we have taken the region of support $u=9$. The experimental results show that this fixed region of support has performed well, even for objects under moderate scale changes. However, when a circular arc is over-enlarged, a
small segment of the arc tends to be flat and the segment in the neighborhood of an inflection point is approximate to a straight line. In contrast, if the circular arc is over-reduced, a point on the arc is similar to the intersection point of two line segments and, therefore, results in high curvature. In such cases, desired feature points may be missed, or spurious corners may be identified. In an additional experiment, we have calculated the included angles of points on arcs and output responses of tangent points and inflection points under different radii ranging from 15 , 20, 25, 30, 35, 40 to 45 pixels. The results shown in Table 5 reveal that the region of support $u=9$ is appropriate to detect tangent points and inflection points for arcs with radii under 35 pixels. For radii larger than 35 pixels, the arc segment within the 9-neighborhood of a point tends to be a straight line, and the output responses of the second ANN are rapidly dropped to zero. Given a boundary containing an arc segment of a specific radius, the maximum included angle among the desired corners must be smaller than the included angles of points on the arc. Otherwise, spurious corners on the arc may be detected.

For images involving multiple arc radii, the region of support for the length of forward (or backward) arm in the first ANN, or for the sign pattern in the second ANN may need to be adaptively determined based on the local properties of the point such as those suggested in $[10,16,26]$. ANN corner detection for objects with multiplesize features is worth further investigation.
(a)
(b)
(c)

Figure 10. Corner detection results for test object 1 (scaling factor 100\%): (a) Desired corners specified by a human viewer. (b) Corners detected by the Freeman-Davis method. (c) Corners detected by the proposed ANN.
(c)

Figure 11. Corner detection results for test object 2 (scaling factor 100\%): (a) Desired corners specified by a human viewer. (b) Corners detected by the Freeman-Davis method. (c) Corners detected by the proposed ANN.
(a)
(b)
(c)

Figure 12. Corner detection results for test object 3 (scaling factor 100\%): (a) Desired corners specified by a human viewer. (b) Corners detected by the Freeman-Davis method. (c) Corners detected by the proposed ANN.


Figure 13. The silhouettes of a real leaf at three scales .
(a)
(b)
(c)

Figure 14. Corner detection results for the leaf (scaling factor 100\%): (a) Desired corners specified by a human viewer. (b) Corners detected by the Freeman-Davis method. (c) Corners detected by the proposed ANN.
(a)
(b)

Figure 15. Test object 4: (a) The binary image of object 4. (b) The detected tangent and inflection points using the second ANN.
(a)
(b)

Figure 16. Test object 5: (a) The binary image of object 5. (b) The detected tangent points using the second ANN.
(c)

Figure 17. Test object 6(a real aluminum casting part): (a) The orignal image. (b) The silhouette of the part. (c) The detected tangent points using the second ANN.

## REFERENCES

1. A. Goshtasby and G. C. Stockman, Point pattern matching using convex hull edges. IEEE Trans. Systems Man Cyber. SMC-15, 631-637 (1985).
2. M. -H. Han and D. Jang, The use of maximum curvature points for the recognition of partially occluded objects, Pattern Recognition 23, 21-33 (1990).

3 H. -C. Liu and M. D. Srinath, Partial shape classification using contour matching in distance transformation, IEEE Trans. Pattern Analysis Mach. Intell. 12, 10721079 (1990).
4. E. R. Davies, Locating objects from their point features using an optimised Hough-like accumulation technique, Pattern Recognition Letters 13, 113-121 (1992).
5. L. Dreschler and H. Nagel, Volumetric model and 3-D trajectory of a moving car derived from monocular TV-frame sequence of a street scene, Proc. IJCAI, 692-697 (1981).
6. M. F. Costabile, C. Guerra and G. G. Pieroni, Matching shapes : a case study in time varying images, Comput. Vision Graphic Image Process. 29, 296-310 (1985).
7. M. Worring and A. W. M. Smeulders, Digital curvature estimation, CVGIP : Image Understanding 58, 366-382 (1993).
8. D. P. Fairney and P. T. Fairney, On the accuracy of point curvature estimators in a discrete environment, Image and Vision Comput. 12, 259-265 (1994).
9. D. -M. Tsai and M. -F Chen, Curve fitting approach for tangent angle and curvature measurements, Pattern Recognition 27,699-711 (1994).
10. C. -H. Teh and R. T. Chin, On the detection of dominant points on digital curves, IEEE Trans. Pattern Analysis Mach. Intell. 11, 859-872 (1989).
11. H. Freeman and L. S. Davis, A corner-finding algorithm for chain-coded curves, IEEE Trans. Comput. C-26, 297-303 (1977).
12. H. L. Beus and S. S. H. Tiu, An improved corner detection algorithm based on chain-coded plane curves, Pattern Recognition 20, 291-296 (1987).
13. G. Medioni and Y. Yasumoto, Corner detection and curve representation using cubic B-splines, Comput. Vision Graphics Image Process. 39, 267-278 (1987).
14. H. -C. Liu and M. D. Srinath, Corner detection from chain-code, Pattern Recognition 23, 51-68 (1990).
15. A. Rattarangsi and R. T. Chin, Scale-based detection of corners of planar curves, IEEE Trans. Pattern Analysis Mach. Intell. 14, 430-449 (1992).
16. B. K. Ray and K. S. Ray, An algorithm for detection of dominant points and polygonal approximation of digitized curves, Pattern Recognition Letters 13, 849856 (1992).
17. C. Fermuller and W. Kropatsch, A syntactic approach to scale-space-based corner description, IEEE Trans. Pattern Analysis Mach. Intell. 16, 748-751 (1994).
18. K. Rangarajan, M. Shah and D. V. Brackle, Optimal corner detection, Comput. Vision Graphics Image Process. 48, 230-245 (1989).
19. R. Mehrotra and S. Nichani, Corner detection, Pattern Recognition 23, 12231233 (1990).
20. L. Kitchen and A. Rosenfeld, Gray level corner detection, Pattern Recognition Letters 1, 95-102 (1982).
21. A. Singh and M. Shneier, Gray level corner detection : a generalization and a robust real time implementation, Comput. Vision Graphics Image Process. 51, 5469 (1990).
22. R. Deriche and G. Giraudon, A computational approach for corner and vertex detection, Int. J. Comput. Vision 10, 101-124 (1993).
23. U. Seeger and R. Seeger, Fast corner detection in gray-level image., Pattern Recognition Letters 15, 669-675 (1994).
24. R. Schalkoff, Pattern Recognition : Statistical, Structural and Neural Approaches, John Wiley \& Sons, New York (1992).
25. D. E. Rumelhart, G. E. Hinton and R. J. Williams, Learning internal representations by error propagation, in Parallel Distributed Processing : Explorations in the Microstructures of Cognition : Foundations, Vol. 1, Eds. D. E. Rumelhart and J. L. McCelland, MIT Press, Cambridge, MA (1986).
26. M. S. Kankanhalli, An adaptive dominant point detection algorithm for digital curves, Pattern Recognition Letters 14, 385-390 (1993).


Fig. 1


Fig. 2


Fig. 3

(a)

(b)

Fig. 3


Fig. 4

(a)

(b)

(c)


(d)


I nput I ayer Hi dden I ayer Out put I ayer

Fig. 6

Table 1(a) Curvature ratios for test object 1 with scaling factor 75\%

| Method | Orientation |  |  |  |  |  | Mean | Std. dev. |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | $0^{\circ}$ | $45^{\circ}$ | $90^{\circ}$ | $135^{\circ}$ | $180^{\circ}$ | $225^{\circ}$ | $270^{\circ}$ | $315^{\circ}$ | $\bar{\Re}$ | $\sigma_{\Re}$ |
| Freeman-Davis | 2.00 | 0.75 | 1.66 | 1.81 | 1.06 | 1.33 | 1.74 | 1.22 | 1.45 | 0.42 |
| ANN | 1.55 | 1.63 | 1.53 | 1.35 | 1.27 | 1.80 | 1.07 | 0.96 | 1.46 | 0.24 |

Table 1(b) Curvature ratios for test object 1 with scaling factor $100 \%$

| Method | Orientation |  |  |  |  |  | Mean | Std. dev. |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | $0^{\circ}$ | $45^{\circ}$ | $90^{\circ}$ | $135^{\circ}$ | $180^{\circ}$ | $225^{\circ}$ | $270^{\circ}$ | $315^{\circ}$ | $\bar{\Re}$ | $\sigma_{\Re}$ |
| Freeman-Davis | 0.97 | 0.22 | 0.48 | 1.02 | 2.44 | 1.10 | 1.02 | 0.94 | 1.02 | 0.65 |
| ANN | 1.46 | 1.41 | 1.30 | 1.15 | 1.59 | 1.34 | 1.51 | 1.55 | 1.38 | 0.18 |

Table 1(c) Curvature ratios for test object 1 with scaling factor $125 \%$

| Method | Orientation |  |  |  |  |  | Mean | Std. dev. |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | $0^{\circ}$ | $45^{\circ}$ | $90^{\circ}$ | $135^{\circ}$ | $180^{\circ}$ | $225^{\circ}$ | $270^{\circ}$ | $315^{\circ}$ | $\bar{\Re}$ | $\sigma_{\Re}$ |
| Freeman-Davis | 2.52 | 0.62 | 1.78 | 1.20 | 0.05 | 1.04 | 1.83 | 0.62 | 1.21 | 0.80 |
| ANN | 1.22 | 1.16 | 1.18 | 1.07 | 1.09 | 1.03 | 1.19 | 2.60 | 1.32 | 0.52 |

Table 2. Average curvature ratios for test object 2 (see Figure 8)

| Method | Scale |  |  | $N_{\Re<1}$ |
| :---: | :---: | :---: | :---: | :---: |
|  | $75 \%$ | $100 \%$ | $125 \%$ |  |
| Freeman-Davis $(\bar{\Re})$ | 0.16 | 0.21 | 0.11 | 24 |
| ANN $(\bar{\Re})$ | 1.03 | 1.12 | 1.03 | 7 |

Note: $\overline{\mathfrak{R}}=$ Average curvature ratios over 8 orientations
$N_{\Re<1}=$ Number of curvature ratios less than 1 for a total of 24 experiments

Table 3. Average curvature ratios for test object 3 (see Figure 9)

| Method | Scale |  |  | $N_{\Re<1}$ |
| :---: | :---: | :---: | :---: | :---: |
|  | $75 \%$ | $100 \%$ | $125 \%$ |  |
| Freeman-Davis $(\overline{\mathfrak{R}})$ | 0.09 | 0.06 | 0.21 | 24 |
| ANN $(\overline{\mathfrak{R}})$ | 1.54 | 1.31 | 1.10 | 2 |

Table 4. Localization of detected corners
(Distances in pixels between true corners and detected corners)

| Scale | Test | Freeman-Davis |  |  |  |  | ANN |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | object | Orientation |  |  |  | Orientation |  |  |  |  |
|  |  | $45^{\circ}$ | $135^{\circ}$ | $225^{\circ}$ | $315^{\circ}$ | $45^{\circ}$ | $135^{\circ}$ | $225^{\circ}$ | $315^{\circ}$ |  |
|  | Object 1 | 1.05 | 1.30 | 1.05 | 1.05 | 0.50 | 0.75 | 0.75 | 0.90 |  |
| $75 \%$ | Object 2 | 1.14 | 0.81 | 0.67 | 1.14 | 0.62 | 0.81 | 0.67 | 0.29 |  |
|  | Object 3 | 2.11 | 1.00 | 2.00 | 1.78 | 0.30 | 1.00 | 0.56 | 0.78 |  |
|  | Object 1 | 0.95 | 1.10 | 1.05 | 0.95 | 0.65 | 0.70 | 0.80 | 0.60 |  |
| $100 \%$ | Object 2 | 1.14 | 0.86 | 1.19 | 1.48 | 0.57 | 0.52 | 0.38 | 0.90 |  |
|  | Object 3 | 2.11 | 1.00 | 2.00 | 1.78 | 0.78 | 1.00 | 0.56 | 1.44 |  |
|  | Object 1 | 0.95 | 0.90 | 1.30 | 1.95 | 0.60 | 0.75 | 1.00 | 1.05 |  |
| $125 \%$ | Object 2 | 2.05 | 1.05 | 1.10 | 1.24 | 1.00 | 0.67 | 0.86 | 0.67 |  |
|  | Object 3 | 2.89 | 2.56 | 1.33 | 1.44 | 1.22 | 1.44 | 1.11 | 0.89 |  |
| Mean distance |  | 1.60 | 1.17 | 1.30 | 1.43 | 0.69 | 0.85 | 0.74 | 0.83 |  |
| Std. dev. |  | 0.62 | 0.48 | 0.46 | 0.42 | 0.25 | 0.24 | 0.23 | 0.29 |  |

Table 5. Effects of different arc radii on included angles of points on arcs and output responses of tangent and inflection points.

| Radius of arc | Average | ANN output responses |  |
| :---: | :---: | :---: | :---: |
| (pixels) | included angle | Tangent point | Inflection point |
| 15 | $156^{\circ}$ | 0.943 | 0.953 |
| 20 | $160^{\circ}$ | 0.956 | 0.944 |
| 25 | $164^{\circ}$ | 0.996 | 0.980 |
| 30 | $167^{\circ}$ | 0.999 | 0.915 |
| 35 | $169^{\circ}$ | 0.998 | 0.532 |
| 40 | $170^{\circ}$ | 0.438 | 0.009 |
| 45 | $171^{\circ}$ | 0.005 | 0.002 |

