

Curvature Estimation and Unique Corner Point Detection for Boundary Representation¹

*Kwanghoon Sohn, Winsor E. Alexander,
Jung H. Kim, Yonghoon Kim, and *Wesley E. Snyder

^{*}Dept. of ECE, NC State Univ., Raleigh, NC 27695-7911, U.S.A.

^{**}Dept. of EE, NC A&T State Univ., Greensboro, NC 27411, U.S.A.

^{***}Bowman Gray School of Medicine, Wake Forest Univ., Winston-Salem, NC 27157, U.S.A.

Abstract

Computing a curvature function on a digitized boundary is an ill-posed problem due to the discrete nature of the boundary. We use a constrained regularization technique to obtain the optimal smooth boundary before computing the curvature function. This method solves a common critical problem of current curvature estimation methods in determining a unique smoothing factor. Ideal corner points rarely exist for a real boundary and may be rounded due to the smoothing effect of the preprocessing. Also, the human cognition system recognizes both ideal corner points and slightly rounded segments as corner points. Thus, we establish a criterion to mimic a human's capability of detecting corner points and to compensate for the smoothing effect of the preprocessing in detecting corner points in the curvature function space.

1. Introduction

It has been noted that the human visual system uses two-dimensional (2-D) boundary information to recognize objects since the pertinent information about an object is contained in the shape of its boundary. There are two main categories in representing objects in general; a *global* method and a *local* method. A global method can not recognize objects that are only partially visible. Thus, many researchers have focused their attention on the local methods, which make recognition possible for the partially visible objects. Since the information of the shape is concentrated at the points having high curvature [1], *corner points* are frequently used for a boundary representation. Hence, computing the curvature function properly is very important for unique corner point detection (or unique boundary representation) and reliable matching.

¹Partially supported by the ARO grant no. DAAL03-90-0913.

The problem of curvature estimation is said to be *ill-posed* because the *existence, uniqueness and stability* of a solution are not guaranteed [2]. The computation of curvature involves the first and second derivatives and differentiating discrete and noisy data enhances the noise. Many curvature estimation methods on a digitized boundary curve have been developed. However, a common critical problem of the existing methods is the difficulty in determining a unique smoothing factor. We solve the above problem by using a *constrained regularization (CR)* technique which combines regularization and a noise equality constraint. The noise equality constraint determines a correct degree of regularization, *i.e.*, a unique smoothing factor. Since the curvature function computed from the optimal smooth boundary has invariant properties for rotation, scale and translation, we can detect corner points invariantly in this curvature function space.

We can detect ideal corner points (singularities in the second derivatives) easily as impulses in the curvature function space. However, ideal corner points rarely exist for a real boundary. They are often rounded due to the smoothing effect of the preprocessing. The human cognitive system recognizes both sharp corner points and slightly rounded segments as corner points. Thus, we establish a criterion, called "*corner sharpness*", qualitatively similar to a human's capability, in which a little tolerance is given to recover slightly rounded corner points.

This paper is structured as follows: In section 2, the existing methods for curvature estimation on a digitized boundary and their problems are discussed. In addition, a new CR approach to optimal boundary smoothing for curvature estimation is introduced. In section 3, a "corner sharpness" is defined for robust corner point detection. Matching results in the presence of occlusion using a 2-D Hopfield neural network are also shown to produce excellent results using this boundary representation. Discussion of the proposed extensions as well as concluding remarks are given in section 4.

2. Constrained regularization approach to optimal boundary smoothing

2.1 Curvature estimation

The curvature is generally defined as the derivative of the tangent angle to the curve. The formula for computing the curvature function using a parametric representation of the curve is obtained as follows:

$$\kappa(t) = \frac{\dot{X}(t)\ddot{Y}(t) - \ddot{X}(t)\dot{Y}(t)}{[\dot{X}^2(t) + \dot{Y}^2(t)]^{3/2}} \quad (1)$$

In a discrete image, the parametrized boundary is represented by an 8-neighbor Freeman chain code, *i.e.*, it is represented by only 8 integer values (0-7), and is therefore severely affected by quantization. Since the formula to compute the curvature function involves the first and second derivatives and the data itself is noisy, the resulting curvature function computed on a digitized boundary is very ragged, and is difficult to use for further processing. The following equation is one of the k -curvature methods which was developed by Groan and Verbeek [3]:

$$\kappa_{ik} = \frac{1}{k} \sum_{j=-k}^{-1} q_{i-j} - \frac{1}{k} \sum_{j=0}^{k-1} q_{i-j} \quad (2)$$

where $q_{i,j}$ is an integer valued (0-7) chain code element. The value of k works as a smoothing factor in Eq. 2. However, it is difficult to determine an optimal (or unique) value of k .

Mokhtarian and Mackworth [4] used Gaussian smoothing to compute curvature at varying levels of detail. They convolved $x(t)$ and $y(t)$ with a 1-D Gaussian kernel of width σ , resulting in a smoothed function $X(t,\sigma)$ and $Y(t,\sigma)$. Hence, rewriting Eq. 1, the discrete curvature using Gaussian smoothing becomes:

$$k(t,\sigma) = \frac{\dot{X}(t,\sigma)\ddot{Y}(t,\sigma) - \ddot{X}(t,\sigma)\dot{Y}(t,\sigma)}{[\dot{X}^2(t,\sigma) + \dot{Y}^2(t,\sigma)]^{3/2}} \quad (3)$$

A 256×256 original gun image was acquired through DT-2851 image acquisition system and the boundary (Fig. 1) was extracted by *contour tracing algorithm* [5]. Fig. 2 shows the results after the Gaussian smoothing for some values of σ . We have noisy estimate of curvature with $\sigma=1$ (Fig. 2(a)), *i.e.*, it is undersmoothed as a result of insufficient filtering. On the other hand, the filter with $\sigma=8$ oversmooths the curvature function (Fig. 2(b)). Thus, the curvature extrema are significantly attenuated, and poorly localized. We can clearly see the difficulty in determining a unique (or optimal) smoothing factor.

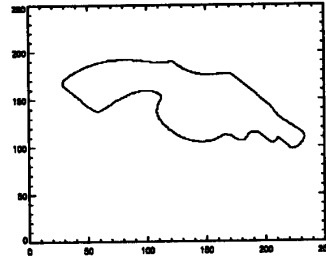
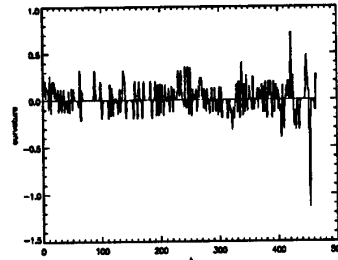
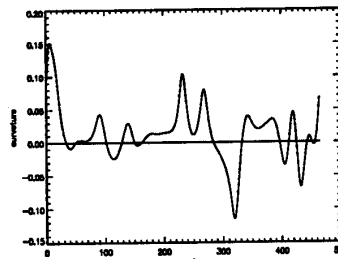


Fig. 1: Gun Boundary.



(a) $\sigma=1$



(b) $\sigma=8$

Fig. 2: Results of Gaussian Smoothing.

2.2 Optimal boundary smoothing for curvature estimation

We assume that an original boundary (f) is not known. It is only known that the function is smooth. The objective of the algorithm is to optimally estimate the original boundary f , given a measured boundary f_m and some knowledge about the noise v ,

$$f_m = f + v \quad (4)$$

We find the optimally estimated smooth boundary (f_s) using CR. The following smoothness requirement is used as a *stabilizing functional* to obtain the smooth curvature estimate since the second derivative represents the roughness of data:

$$\min_t \int [f_s''(u)]^2 du \quad (5)$$

Using a discrete form, we summarize the problem as, **Problem Statement:** Find a boundary vector f_s to minimize the following stabilizing functional:

$$\min \left\{ \sum_k [f_s(k+1) - 2f_s(k) + f_s(k-1)]^2 \right\} \quad (6)$$

subject to the following noise equality constraint,

$$(f_m - f_s)^T (f_m - f_s) = f_m^T f_m = \epsilon \approx N\sigma_n^2 \quad (7)$$

where N is a number of data points, f_m is a residual vector, ϵ is a residual error and σ_n^2 is a noise variance. We do not know the noise function, but we assume we know some of its statistical properties. In other words, if $v(k)$ has zero mean, then

$$v^T v = \sum_k v^2(k) = N\sigma_n^2 \quad (8)$$

Thus, we find f_s such that the residual f_m has the same statistical property as the noise vector (v).

The above problem statement can be handled without difficulty by using the method of *Lagrange multipliers* as follows:

$$J(f_s, \lambda) = f_s^T C_s^T C_s f_s + \lambda [(f_m - f_s)^T (f_m - f_s) - \epsilon] \quad (9)$$

where λ is a Lagrange multiplier. Since the boundary is closed, C_s is an $N \times N$ circulant matrix,

$$C_s = \begin{bmatrix} -2 & 1 & 0 & \dots & 0 & 0 & 1 \\ 1 & -2 & 1 & \dots & 0 & 0 & 0 \\ & & & \vdots & & & \\ 0 & 0 & 0 & \dots & 1 & -2 & 1 \\ 1 & 0 & 0 & \dots & 0 & 1 & -2 \end{bmatrix} \quad (10)$$

Differentiating Eq. 9 with respect to f_s and λ yields

$$\begin{aligned} (f_m - f_s)^T (f_m - f_s) &= \epsilon \approx N\sigma_n^2 \\ f_s &= \left(I + \frac{1}{\lambda} C_s^T C_s \right)^{-1} f_m = (I + \gamma C_s^T C_s)^{-1} f_m \end{aligned} \quad (11)$$

where $\gamma = 1/\lambda$ and I is an $N \times N$ identity matrix.

Since the size of the matrix used in Eq. 11 is $N \times N$ and N is usually large, it is clear that the inversion of C_s is impractical. This difficulty can be avoided by making use of the properties of circulant matrices [6]. That is, we can estimate the boundary in a frequency domain using Discrete Fourier Transform (DFT) as follows:

$$F_s(u) = \frac{1}{1 + \gamma(D_c)_u} F_m(u) \quad (12)$$

where $F_s(u)$ and $F_m(u)$ are DFTs of $f_s(k)$ and $f_m(k)$, and D_c is the diagonalized matrix of $C_s^T C_s$. Finally, we obtain the optimally estimated boundary vector f_s by taking the inverse DFT of Eq. 12. The optimal value γ^* is iteratively obtained according to the constraint

(Eq. 7). The CR technique is a data-driven method. In other words, the level of noise of the boundary data determines the amount of smoothing. We can obtain the smooth curvature function from the above optimal smooth boundary by Eq. 1.

2.3 Experimental results

Fig. 3(a) shows a smooth boundary generated in [7]. The original boundary was quantized to test the performance of the algorithm. Fig. 3(b) shows the noisy curvature function of the quantized boundary computed by Eq. 1. The algorithm was applied to the x - and y -coordinates separately. The optimal values for the quantized boundary were $\gamma_x^* = 1069$ and $\gamma_y^* = 2900$. Then, the curvature function is computed using the smoothed x - and y -coordinate values according to Eq. 1. Fig. 3(c) shows the excellent results of the algorithm. Since each interval (t) has different arclength due to the effect of the Freeman chain code, we should make them uniform about the arclength for further processing. A *linear interpolation* was used for this purpose.

In order to see the effect of *scale* factor, we tested the algorithm for a half-scaled boundary. The optimal values, $\gamma_x^* = 1585$ and $\gamma_y^* = 5457$, were obtained for this boundary. The resulting curvature function after the algorithm was applied was very smooth and had the twice larger curvature value than the original one. We found that the curvature functions are inversely proportional to the scale factor. However, the *scale* invariance can be remedied by normalizing the curvature function. We rotated and translated the boundary, and then quantized to see the rotation and translation invariance properties. We applied the algorithm to this boundary and obtained $\gamma_x^* = 897$ and $\gamma_y^* = 1643$. We obtained exactly the same result as the smooth curvature in Fig. 3(c).

The new method discussed in this section avoids the difficulty of determining a unique smoothing factor in other methods by smoothing the proper amount of noise according to the noise equality constraint. Another advantage is that we can significantly reduce the computation time by using the properties of circulant matrices. In addition, the algorithm preserves the useful invariance properties of the curvature function. However, the algorithm smooths out sharp corner points to some extent, even though this smoothing is not serious.

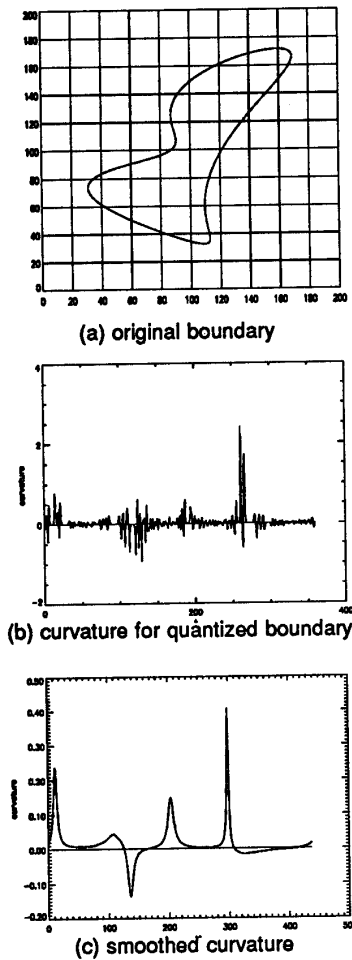


Fig. 3: Results of CR.

3. Unique boundary representation

A sharp corner point may be slightly smoothed as a result of the regularization. Thus, we may lose some corner points due to this unnecessary smoothing effect on corner points. We overcome this problem by defining "corner sharpness" in this section.

3.1 Robust corner point detection

A human recognizes slightly rounded segments as corner points as well as sharp corner points. Thus, we propose a new method to mimic a human's capability of detecting corner points based on the definition of "corner sharpness". Fig. 4 shows a set of corner points

which are basically in the same shape. Notice that '*' indicates an ideal corner point, a rounded corner whose arclength is ds_2 appears lower and wider in the curvature function space, while less rounded corner whose arclength is ds_1 has higher and narrower curvature. If we use only a single threshold value for detecting corner points in the curvature function space, we may miss rounded corners with a wider and shorter curvatures (κ_1 and κ_2). The criterion for detecting a corner point in this continuous interpretation is more related to cognitive science, and may be a fuzzy factor in the human cognition process. However, for machine recognition, we can overcome the above problem by establishing a criterion of corner sharpness. As the corner point is rounded for some reason, the corresponding curvature function is spread over the neighbor, while the overall change of tangent angle around the corner point remains the same, i.e., $ds_1 \times \kappa_1 = ds_2 \times \kappa_2 = \theta$ (Fig. 4(b)). When the corner point is spread over a tolerably narrow number of neighbors, it still can be recognized as a corner point. However, when the corner is rounded too much, it should be recognized as a curved segment instead of a corner point.

We define "corner sharpness" to cover all the above cases.

Definition 3.1 (Corner sharpness): Corner sharpness is defined as a tangent angle change ($\Delta\theta$, in a normalized curvature function space) within a unit perimeter (Δs) of a simply closed boundary. A point on the boundary is considered as a *corner point* if $\Delta\theta$ within Δs around the point satisfies the following condition:

$$\Delta\theta = \int \kappa ds > \theta_{th}$$

where θ_{th} is a threshold value.

Since the normalized curvature function is uniquely obtained, we can detect corner points uniquely according to the corner sharpness at each point of a boundary. We determine a threshold value (θ_{th}) for the corner point detection in the normalized curvature function. It is based on a tangent angle change within a unit perimeter length of a circle whose perimeter is 1 as follows. Assume that the boundary is sampled to 360 points with equal arclengths. Thus, the unit perimeter length (Δs) of the circle is $1/360$ and the tangent angle change ($\Delta\theta$) within Δs is 2π . We consider a point as a corner point in a simply closed boundary if the corner sharpness within a unit perimeter is 10 times greater than the above $\Delta\theta$ (2π). Since the normalized curvature function value at each point for the circle is 2π , the threshold value, $\theta_{th} = 20\pi$, is obtained. However, the CR method smooths out corner points slightly and the resulting curvature at corner points are spread over

the neighbors. Thus, we detect corner points based on the following empirically derived result when we use the CR method:

Each point on a boundary is considered as a corner point if one of the following conditions are satisfied:

$$\int_0^{c\Delta s} \kappa ds \geq 10\pi(c+1) \quad , \text{ for } c=1, 2, 3 \quad (14)$$

where we define $c=1,2,3$ as stage-1, stage-2, and stage-3. Fig. 5 shows a little tolerance for robust corner point detection when we use the CR method. Only the points in the shaded region of the graph, which satisfy one of the above stages, are considered as corner points. The points in the black region are also above the line. However, they are recognized as curved segments instead of corner points since the corner is too rounded.

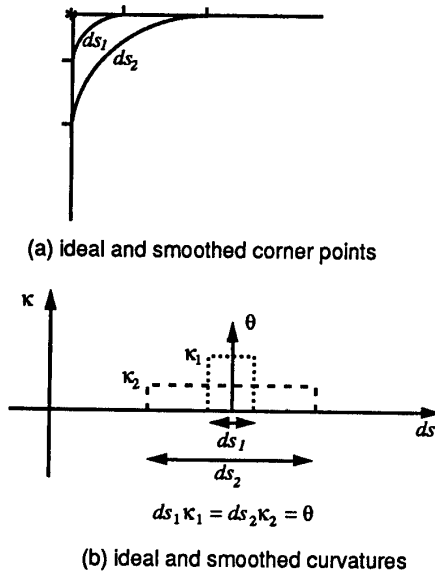


Fig. 4: Smoothing effect of corner points.

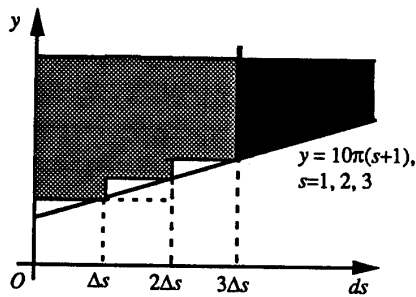


Fig. 5: Practical region of corner points.

3.2 Experimental results

Fig. 6(a) shows the overall corner points detected by corner sharpness for a gun boundary (Model-1). A few corner points were not detected at stage-1 and stage-2 due to the unnecessary smoothing effect on corner points. However, the missed corner points were recovered at stage-3. Fig. 6(b,c) are other examples of corner points detected by corner sharpness for a hammer (Model-2) and an occluded object (Input-1) between Model-1 and Model-2. We can see the good performance of the algorithm. As a demonstration of the power of the corner point detection algorithm, we consider an example of matching the silhouette of a partially occluded object to a model. Any number of matching methods could be used for this demonstration; here we formulate object recognition as matching a model graph with an input image graph. We pose this graph matching problem as an optimization problem where an energy function is minimized. This optimization problem can be mapped into a 2-D Hopfield network [8].

We used a continuous type of 2-D Hopfield network for matching. A graph is constructed to create a model for each object using unique corner points as nodes of the graph. Each node has local features (angle) as well as relational features (distances between nodes) with other nodes. Not only relations between neighboring corner points but also relations between all other corner points were used as constraints to increase the robustness of the algorithm. During matching procedure, a similar graph is constructed for the input image which may consist of one or several occluded objects. Each model graph is then matched against the input image graph to find the best matching subgraph. Fig. 7 shows the matching results. The nodes marked '*' in the figure represent the matched nodes. It shows the excellent matching results between a model object and an input object in the presence of occlusion.

4. Conclusion

Issues related to the unique boundary representation for a reliable matching have been discussed in this paper. The current methods to compute curvature on a digitized boundary have a common difficulty in determining a unique smoothing factor. We solved this problem by applying a constrained regularization technique to the digitized boundary. By using properties of circulant matrices we significantly reduced the computation time since we avoided inversion of a large matrix. From the smooth boundary, we obtained a

unique curvature function which had invariant properties. Thus, we could detect corner points invariantly. We defined corner sharpness to compensate for the slight smoothing effect of the regularization as well as to mimic a human's behavior of detecting corner points. Matching results based on the above unique boundary representation support the excellence of our methods, even in the presence of occlusion.

5. References

- [1] F. Attneave, "Some informational aspects of visual perception," *Psychol. Rev.*, vol. 61, no. 3, pp. 183-193, 1954.
- [2] J. Hadamard, *Lectures on the Cauchy problem in linear partial differential equations*, Yale University Press, 1923.
- [3] F. C. Groan and P. W. Verbeek, "Freeman-code probabilities of object boundaries quantized contours," *Comput. Vision, Graphics, Image Processing*, vol. 7, pp. 391-402, 1978.
- [4] F. Mokhtarian and A. Mackworth, "Scale-based description and recognition of planar curves and two-dimensional shapes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, no. 1, Jan. 1986.
- [5] T. Pavlidis, *Algorithms for Graphics and Image Processing*, Murray Hill, New Jersey, Computer Science Press, 1981.
- [6] B. Hunt, "The application of constrained least squares estimation of image restoration by digital computer," *IEEE Trans. Comput.*, vol. C-22, no. 9, pp. 805-812, Sep. 1973.
- [7] B. Shahraray and D. J. Anderson, "Optimal estimation of contour properties by cross-validated regularization," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-11, no. 6, pp. 600-610, June, 1989.
- [8] W. C. Lin, F. Liao, C. Tsao, and T. Lingutla, "A hierarchical multiple-view approach to three-dimensional object recognition," *IEEE Trans. Neural Networks*, vol. 2, no. 1, pp. 84-92, Jan. 1991.

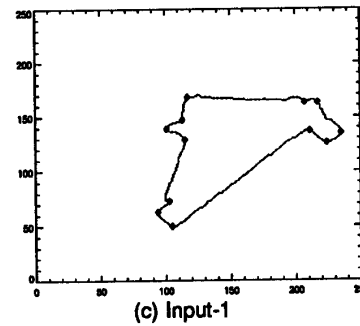
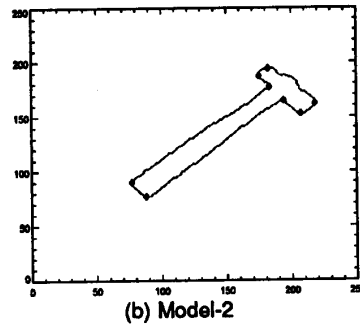
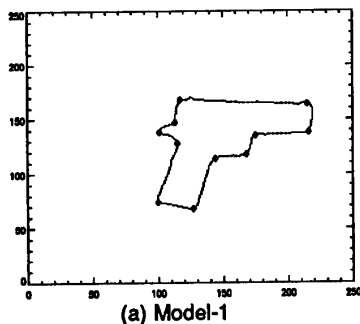


Fig. 6: Corner Points by Corner Sharpness.

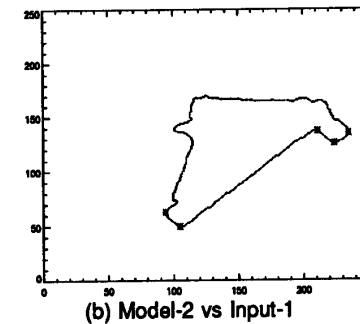
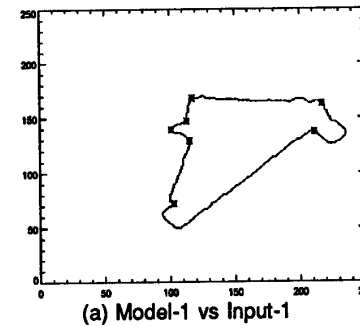


Fig. 7: Matching Results.