Joint Conference of 1996 International Computer Symposium
December 19~21, Kaohsiung, Taiwan, R.O.C.

The Neck-based Parts Computation Method for Shape Decomposition

Kridanto Surendro and Yuichiro Anzai

Department of Computer Science, Keio University
Yokohama 223, Japan
{endro,anzai}@aa.cs.keio.ac.jp

Abstract

In the task of forming high-level object-centered models from low-level image-based features, parts serve as an intermediate representation. A representation of parts for object recognition should be rich, stable, and invariant to changes in the viewing conditions. In addition, it should be capable of describing partially occluded shapes. This paper describes a method for decomposing a shape into its parts. The method is based on the negative curvature minima which have co-circular tangents, and directly compute necks from limb-based computation with a certain condition. Therefore, the method is efficient for partitioning a shape into the parts. Currently, we assume that the shape is a closed planar curve.

Keywords: Vision, Recognition, Shape Decomposition, Parts

1 Introduction

In the absence of distinguishing properties such as color, texture, or motion, object recognition first requires the visual recovery of shape [5]. Here, part-based representations may be the key to the general applicability of invariant representations to object recognition [14], and allow for recognition that is robust in the presence of occlusion, movement, deletion, or growth of portions of an object.

Many researchers have suggested a representation scheme based either on the decomposition of its interior region or on the contour-based segmentation of its boundary. Region-based segmentations include: parts obtained at branches of the symmetric axis transform [8]; parts obtained by finding protrusion-like regions that are as close as possible to being convex [13]; parts obtained by finding the best combinations of a set of volumetric primitives such as generalized cylinders [6] or superquadrics [1]. In contrast, contour-based segmentations use boundary features such as high-curvature points [9].

Following psychophysical observations, Hoffman and Richards [3, 19], who advocate that part decomposition should precede part description, suggest that shapes are naturally segmented into parts at negative curvature minima. Leyton [15, 16] has suggested a process grammar for shape. He argues that a shape can be understood as the outcome of processes (deformations) that formed it. Based on an important duality between curvature extrema and symmetry, he uses symmetry axes to infer process history.

To merge these extreme views, Kimia et al. [2] propose a continuum connecting two extremes, and capturing the distinction between, (1) the parts extreme where objects have clearly defined and distinct segments, e.g., resulting from object composition, and (2) the protrusions extreme, where an object is best described as another whose boundary has been deformed, e.g. due to growth. They propose a theory of shape based on a reaction-diffusion equation. Their method produces a hierarchical decomposition of shape into parts and protrusions, which are defined as singularities in the continuous reaction-diffusion space.

Currently, using both region and contour information, Rom and Medioni [10] propose an approach by using smooth local symmetries (SLS's) for the axial description of parts and parallel symmetries to provide information on global relationships within the shape. While Siddiqi and Kimia [11] suggested a method for computing parts based on limbs and necks partitioning scheme. The method produces a two kinds of parts, those that are limb-based and those that are neck-based. But the computational approach for this scheme resulting a binary computation, thus make it inefficient.

This paper discuss the new approach for partitioning necks by direct computation.

2 Overview of the Approach

We suggested an approach for computing neck directly from limb-based computation, therefore avoid the binary part computations. Also, instead of assuming the availability of an ordered list of the shape's contour points, we are approximate the input shape with approximating B-splines. We outline our method here, and the details of our implementation are presented in the next section. Fig. 1 presents an overview of our approach.

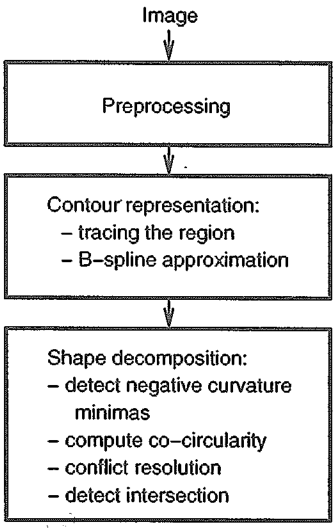

Fig. 1. Overview of the approach.

Initially we perform some preprocessing on the image to obtain a binary of the input shapes, and approximate it with approximating B-splines. Following this step, we have an analytical representation of the contour.

The negative curvature minimas are detecting and rule out the curvature extrema that do not represent significant changes in orientation. Candidate limbs are extracted by computing co-circularity between each pair of negative curvature minima. Conflict between candidates are resolved by computing a salience for each.

We propose a new method to obtain neck as following. First, detect the existence of an intersecting part-line that connect between a pair of negative curvature minima, find the orientation of each line, and compute the angle between it. Second, compare the angle obtained with an upper treshold. Neck is found, if the magnitude of the angle is below an upper treshold.

3 Shape Representation

The B-splines are use to represent the shape of the input planar curves. The B-splines are piecewise polynomial functions that provide local approximation to contours using a small number of parameters (control points). They have been extensively used in the representations of curve and surfaces in computer graphics [4], and shape representation [7, 18].

The following attractive properties make this approximation suitable for shape representation and analysis:

- The B-spline is characterized by a set of parameters (the control points) from which the curve can be totally generated.
- Smoothness and continuity, which allows any curve to consist of a concatenation of curve segments, yet be treated as a single unit.
- Shape invariance under transformation(affine and projective transformations), which means that the projected/transformed curve is still a B-spline.
- Local controllability, which implies that local changes in shape are confined to the B-spline parameters local to that change.

We use uniform cubic B-spline, that have continuous first and second derivatives everywhere, to approximate the shape of the image curves. We will not present the formal mathematical definition here, and suggested the readers to see [4].

4 The Shape Partitioning Schema

The limb and neck partitioning schema was proposed by Siddiqi [11]. Hence, recovering parts rely both on properties of objects and on the nature of visual projection. The notion of necks and limbs were developed as a result of an interaction among psychopysical, "ecological" and computational considerations [12].

Essentially, shapes are to be partitioned at limbs, or pairs of negative curvature minima whose boundaries show evidence for good continuation, and necks, or locally thinnest regions (see Fig.2 and Fig.3). A part-line is a curve that divides a shape into parts at limbs or necks.

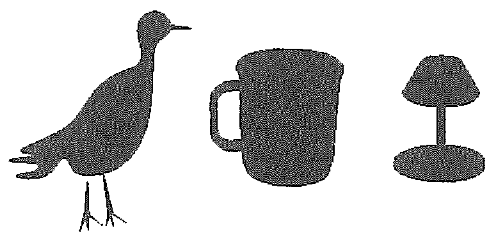

Fig. 2. Examples of limb-based parts.

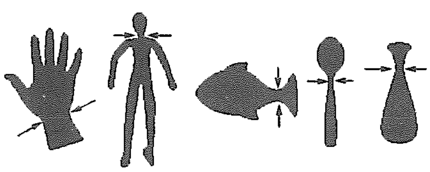

Fig. 3. Examples of neck-based parts.

The algorithm for computing limbs is following.

- **Detecting negative curvature minima.** We use the first derivative of the B-spline to detect the negative curvature minima and then calculate the local extrema in curvature. Then make a part-line, that is a pair of the detected negative curvature minima with local extrema in curvature.

- **Extracting candidate limbs.** The good continuation of each pair of negative curvature minimas are obtained by computing the co-circularity of each pair. Two unit tangents λ and λ' are co-circular, denoted $\lambda \asymp \lambda'$, iff there exists a circle to which they are both tangent.

The following is computation for cocircularity (see [17], for detailed). Let (x_i,y_i) and (x_j,y_j) be the coodinates of nodes i and j, and let (x, y) be an arbitrary point within the circle of radius $\frac{1}{2}$ centered at $(x_i,\ y_i)$, and $(x',\ y')$ a point in a circular neighborhood of $(x_j,\ y_j)$; let λ and λ' be unit tangents at these locations and θ_λ and $\theta_{\lambda'}$ be their respective orientations; let θ be an orientation in an ϵ-neighborhood of θ_λ, and $\theta_{\lambda'}$ an orientation in an ϵ-neighborhood of $\theta_{\lambda'}$ (see Fig. 4).

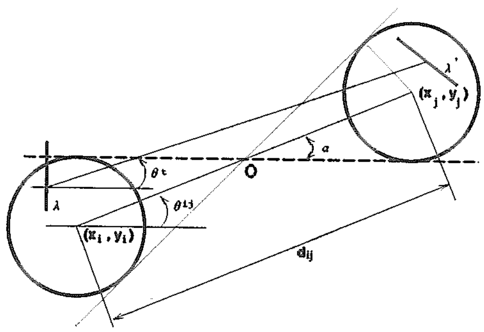

Fig. 4. Two unit tangents λ and λ' with respective orientations θ_λ and $\theta_{\lambda'}$ whose positions are restricted to the circle of radius $\frac{1}{2}$ pixel.

The orientation of the line joining the centers of the pixels is given by

$$\theta_{ij} \;=\; \arctan(\Delta y/\Delta x) \qquad (1)$$

where $\Delta x = x_j - x_i$ and $\Delta y = y_j - y_i$. In the case of circles of equal radii, the angle between the common tangent and the line joining their centers is given by

$$\alpha \;=\; \arcsin(\frac{1}{d_{ij}}) \qquad (2)$$

Let the function $\Gamma(\beta,\ \gamma)$ designate the interior angle between a pair of lines with orientations β and γ. The tangents at $(x_i,\ y_i)$ and $(x_j,\ y_j)$ are cocircular if the interior angle between the first tangent and the line joining the tangents is the same as that between the latter line and the second tangent. Formally, $\lambda \asymp \lambda'$ iff

$$\Gamma(\theta,\theta_t) \;=\; \Gamma(\theta_t,\theta'), \qquad (3)$$

for some $\theta_t \in (\theta_{ij}-\alpha,\ \theta_{ij}+\alpha)$, $\theta \in (\theta_\lambda-\epsilon/2,\ \theta_\lambda+\epsilon/2)$, $\theta' \in (\theta_{\lambda'}-\epsilon/2,\ \theta_{\lambda'}+\epsilon/2)$. This condition is clearly equivalent to

$$|\Gamma(\theta_\lambda,\theta_{ij}) - \Gamma(\theta_{ij},\theta_{\lambda'})| \;<\; \epsilon + 2\alpha \qquad (4)$$

- **Conflict resolution.** Conflicts between candidate limbs are resolved by computing total curvature and extent. Consider the co-circular boundary tangents T_l^1, T_l^2 which form angles θ_l^1, θ_l^2 respectively (see Fig. 5).

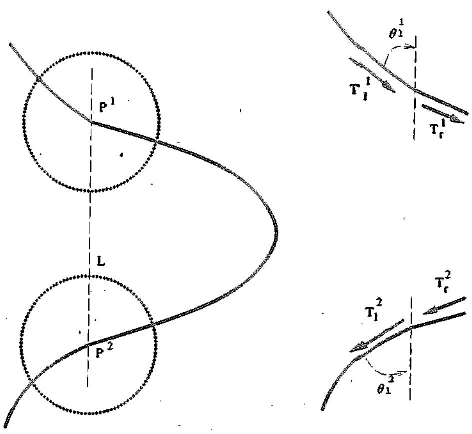

Fig. 5. The left and right neighborhoods at each endpoint of the part-line.

Total curvature is calculated as

$$\frac{\theta_l^1 + \theta_l^2}{\pi}. \qquad (5)$$

Extent is computed as the ratio of major to minor axes of the largest ellipse that can be inscribed in the shape, but restricted to lie within local neighborhood. Extent attains a maximum when the major axis of the inscribed ellipse coincides with the diameter of the local neighborhood's disk, and is calculated as (extent/2α), where α is a constant. Finally compute the average of

$$(1 - \frac{\theta_l^1 + \theta_l^2}{\pi}) + \frac{extent}{2\alpha}. \qquad (6)$$

The neck-based parts are computed by finding all locally shortest part-line between two points on the boundary, for which a circle can be inscribed in the shape, and selecting those whose curvature disparity exceeds a low threshold. This brute force tehnique for computing neck is computationally expensive.

As proved by [11], the method is invariant, robust, stable and yield a hierarchy of parts. But the computational approaches are binary: in that a part-line is either a neck or not; is either a limb or not. This condition lead us to suppose a non-binary part computation that will be describe below.

5 The Neck Computation

Consider the tail of the fish on Fig. 6, that was recognized as a neck-based part of the fish shape.

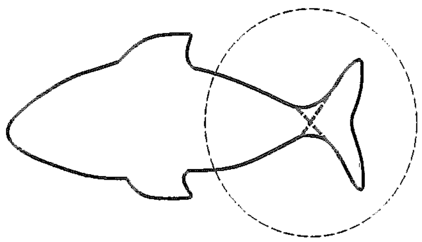

Fig. 6. The fish tail that was recognized as the neck-based part.

When computing limbs, we found candidate necks are always signaled by the intersection of the part-lines. This condition means that necks are always found between two parts, as suggested by [2] that necks emerge as intersecting protrusions connecting coupled parts.

In this paper we propose a method to recognize the candidate necks as follows. The existence of necks are obtained when there is an intersecting between the lines of candidate limbs. It is mean that necks are directly obtained from the computation of limbs by adding a condition for the intersecting lines. Therefore the computation is efficient.

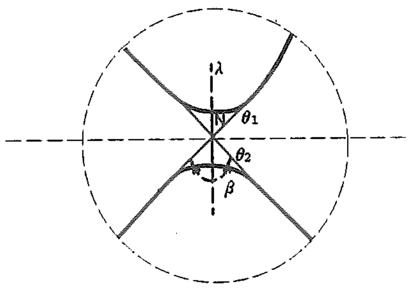

Fig. 7. θ_1 and θ_2 are orientation of the part-line of the candidate limbs that have been found, N is the part-line of the neck with orientation λ.

Necks are extracted (see Fig. 7) by 1) detecting the intersection line of candidate limbs, 2) computing the orientation of each line and compute the angle, β between it, 3) comparing β with the upper treshold, neck is obtained when the angle is below than the treshold, and 4) computing the orientation of the part-line for neck-based part. If the intersection of part-lines are found, with orientation θ_1 and θ_2, the angle between the two orientation is

$$\beta = |\theta_1 - \theta_2|. \quad (7)$$

Comparing β with upper treshold, the neck-based part is found when β less then the treshold. The orientation of part-line, λ, for the neck-based part is approached by computing

$$\lambda = \frac{\beta}{2}. \quad (8)$$

6 Examples

We illustrate our method on several examples of man-made and biological objects using the same parameter values. For man-made objects, there are 2 candidate neck-based resulted from our proposed method on Fig. 8. left, and one candidate neck-base for the shape on Fig. 8. right. If the intersection is found, then calculate the angle between two intersecting lines to determine the orientation of the neck-based part-line.

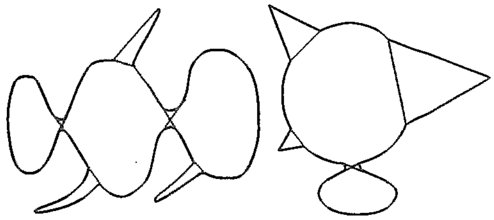

Fig. 8. The man-made objects shape.

Decomposing the shape of biological objects (Fig. 9) give an intuitive result, where animal-like's shape are partitioned on head, legs, etc.

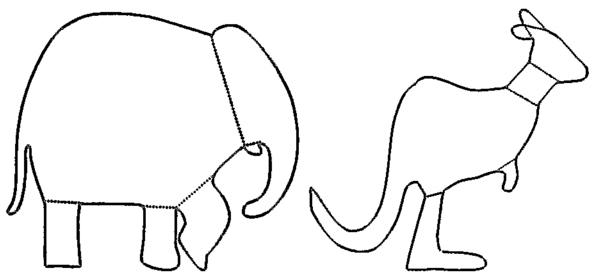

Fig. 9. The biological objects shape.

Decomposing an example of the plane, give a result that can be seen on Fig. 10.

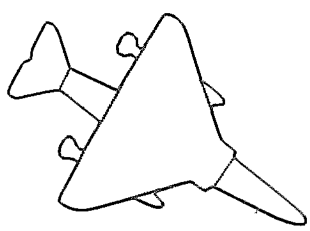

Fig. 10. an example of the plane shape.

In general, the results are similar with [11], and also on we could not recognize the trunks and tails as parts of the shape, as can be seen on Fig. 9.

7 Concluding Remarks

We have presented a method for obtaining natural descriptions of planar shapes, that directly computes neck from limbs computation. Our method is computationally efficient, robust to changes in parameter settings, and stable. The results show that it provides intuitive shape descriptions for various shapes.

References

[1] A. Pentland. "Recognition by parts," *First International Conf. on Computer Vision,* (London, England, June 8-11, 1987), Washington, DC., 1987, IEEE Computer Society Press.

[2] B.B. Kimia, A.R. Tannenbaum, and S.W. Zucker, "Toward a computational theory of shape: an overview," O.Faugeras, editor, Lecture Notes in Computer Science, 427, pages 402-407, Berlin, 1990, Springer Verlag.

[3] D.D. Hoffman, and W.A. Richards, "Parts of recognition," M.A. Fischler, editor, *Readings in Computer Vision,* pp.227-242, Morgan Kaufmann Publisher, Inc. 1987.

[4] D.F. Rogers, and J.A. Adams, *Mathematical Elements for Computer Graphics,* 2nd ed. McGraw-Hill International Editions,1990.

[5] D. Marr, *Vision,* San-Fransisco, CA: W.H. Freeman,1982.

[6] D. Marr, and K.H. Nishihara, "Representation and recognition of the spatial organization of three dimensional structure," *Proceedings of the Royal Society of London,* B 200:269-294, 1978.

[7] F.S. Cohen, and J.Y. Wang, "Part I: Modeling image curves using invariant 3-D object curve models -a path to 3-D recognition and shape estimation from image contours," *IEEE Transaction on Pattern Analysis and Machine Intelligence,* Vol.16, No.1, January 1994.

[8] H. Blum,and R.N. Nagel, "Shape description using weighted symmetric axis features," *Pattern Recognition,* Vol.10, pages 167-180, 1978.

[9] H. Freeman, "Shape description via the use of critical points," *Pattern Recognition,* Vol.10, pages 159-166, 1978.

[10] H. Rom, and G. Medioni, "Hierarchical Decomposition and Axial Shape Description," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol.15, No.10, October 1993.

[11] K. Siddiqi, and B.B. Kimia, "Parts of visual form: Computational Aspects," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol.17, No.3, March 1995.

[12] K. Siddiqi,K.J. Tresness, and B.B. Kimia, "Parts of visual form: Ecological and psychophysical aspects," Technical Report LEMS 104, LEMS, Brown University, June 1992.

[13] L. Shapiro, and R. Haralick, "Decomposition of two-dimensional shapes by graph-theoritic clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 1:10-20, 1979.

[14] M. Hebert, J. Ponce, T. Boult, and A. Gross, "Report on the 1995 Workshop on 3-D Object Representations in Computer Vision," M.Hebert et al, editor, *Lecture Notes in Computer Science,* 994, pages 1-18, Berlin, 1995, Springer Verlag.

[15] M. Leyton, "Symmetry-curvature duality," *Computer Vision Graphics Image Processing,* vol.38, pages 327-341, 1987.

[16] M. Leyton, "A process-grammar for shape," *Artificial Intelligence,* Vol.34, pages 213-247,1988.

[17] P. Parent,and S.W. Zucker, "Trace inference, curvature consistency, and curve detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol.11, No.8, August 1989.

[18] P. Saint-Marc, H. Rom, and G. Medioni, "B-Spline contour representation and symmetry detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol.15, No.11, November 1993.

[19] W. Richards,and D.D. Hoffman, "Codon Constraints on Closed 2D Shapes", M.A. Fischler, editor, *Readings in Computer Vision,* pp.700-708, Morgan Kaufmann Publisher, Inc. 1987.