Segmenting by Seeking the Symmetry Axis

Tyng-Luh Liu
IIS, Academia Sinica
Nankang, Taipei 115, Taiwan
liutyng@iis.sinica.edu.tw

Davi Geiger
Courant Institute, NYU
New York, NY 10012, USA
geiger@cs.nyu.edu

Alan L. Yuille
Smith-Kettlewell Institute
San Francisco, CA 94115, USA
yuille@skivs.ski.org

Abstract

We introduce a method for segmenting a shape from an image and simultaneously determining its symmetry axis. The symmetry is used to help the segmentation and in turn the segmentation determines the symmetry. The problem is formulated as one of minimizing a goodness of fitness function and Dijkstra's algorithm is used to find the global minimum of the cost function. The results are illustrated on real images.

1. Introduction

The importance of symmetric shapes has long been realized in the vision community [2, 3, 4]. Such symmetries, when detected, contain useful information about the shape which can be used for shape description and for breaking shapes up into symmetric parts. In addition, it is clear that human observers are highly sensitive to symmetry and make use of it in their processing of images [21].

Most work on symmetry has assumed that the shape has first been segmented from the background. In practice, however, image segmentation is often very difficult to do correctly. It is therefore unrealistic to assume that current general purpose segmentation techniques will be able to determine the shapes' silhouette. In addition, such techniques will often output large numbers of edges in the image and a grouping process will be needed to determine which edges correspond to the shape and which do not.

This motivates us to devise a method to segment the shape and determine its symmetry axis *simultaneously*. In this method, the symmetry helps the segmentation process and, in turn, the segmentation guides the symmetry detection. Our method involves defining a goodness of fitness function, which embodies our notion of symmetry, and of using the shortest path algorithm (i.e., Dijkstra's algorithm, an A^* like algorithm that uses the best-first search principle) to find a global minimum of this function.

2. Previous Work

There have been many papers on symmetry detection. In [2, 3, 4] Blum proposed the *symmetry axis* (or *skeleton*) as a shape description for objects. A radius function and a curvature function are defined for points on the symmetry axes. These functions together with the symmetric axes can then be used as a shape descriptor. More recent work has included variants such as SLS [5] and PISA [13]. Moreover, much progress has been made applying multiresolution schemes and/or smoothing [17], ([16], via Voronoi diagrams), [18], ([19], via reaction-diffusion equation), [22]. In [14], a variational framework based on self-similarity of shapes is presented.

These methods, however, assume that the silhouette of the shape has first been extracted. But image segmentation is a difficult problem and, in addition, edge grouping would be required in order to determine which edges correspond to the object's silhouette. For color images it is possible to segment using regional cues [11], but these methods have not yet been extended to grey-scale images where regional properties are typically less informative. More relevant is the work on finding roads by dynamic programming [1].

On the other hand, there has been success using high level object models to perform segmentation of flexible objects [12, 10, 8]. These methods, however, are too object specific for our purposes. We require a method that uses more knowledge than a typical low level segmentation algorithm but less than an object specific model. Symmetry axes are a good intermediate level representation and their

output could be used as input to shape representation and recognition systems such as [19, 22, 14]. Ad hoc approach uses symmetry axis can be found in [7] but here we address it in a principle way.

Our approach will use Dijkstra's algorithm, where in object recognition is introduced in [10]. This algorithm is related to dynamic programming which has been used in several vision applications, for example [15, 1, 9, 8]. Dijkstra's algorithm is guaranteed to find the optimal solution and is typically considerably faster than dynamic programming.

3. The Representation of Symmetry and Shape

Our approach involves first defining a representation model for the shape and its symmetry axis. Next we define a goodness of fitness function which measure how well the model fits the data. Finally we describe how Dijkstra's algorithm can be used to minimize the goodness of fitness function.

We represent the shape via a *symmetry axis* as follows. For each point $\vec{a}_i$ in the symmetry axis we define a pair of corresponding points $(\vec{p}_i^{\,l}, \vec{p}_i^{\,r})$ on the *left and right boundaries* of the shape, see figure (1). Moreover, $a_i = \frac{1}{2}(\vec{p}_i^{\,l} + \vec{p}_i^{\,r})$ so that the symmetry axis is defined by the corresponding pair of boundary points. We define *rib vectors* $\{\vec{R}_i\}$ so that $\vec{R}_i = (\vec{p}_i^{\,r} - \vec{a}_i) = -(\vec{p}_i^{\,l} - \vec{a}_i) = \frac{1}{2}(\vec{p}_i^{\,r} - \vec{p}_i^{\,l})$.

We can choose to describe the shape in terms of either the left and right boundaries or the symmetry axis and the rib vectors. In this paper we use the left and right boundaries as the fundamental representation of the shape. The symmetry axis and the rib vectors are then treated as derived quantities which can be treated as functions of the boundaries.

Another derived quantity which we will also use is the *rib angles* $\{\theta_i^r\}$ and $\{\theta_i^l\}$, where $\cos\theta_i^r = (\vec{a}_i - \widehat{\vec{a}_{i+1}}) \cdot \widehat{\vec{R}_i}$ and $\cos\theta_i^l = -(\vec{a}_i - \widehat{\vec{a}_{i+1}}) \cdot \widehat{\vec{R}_i}$. We use $\widehat{\ }$ to denote the unit vector. Observe that $\theta_i^r + \theta_i^l = \pi$ for each i.

Finally, we define the *curvature angles* $\{\phi_i\}$. These angles are a measure of straightness of the symmetry axis. They are defined by $\cos\phi_i = (\widehat{\vec{a}_{i+1} - \vec{a}_i}) \cdot (\widehat{\vec{a}_i - \vec{a}_{i-1}})$.

4. The Goodness of Fitness Function

We now define a goodness of fitness function to determine how well the shape model fits the data. The fitness function must satisfy several desirable properties. The first property is that it must ensure that the boundaries of the shape should generally be at places of high intensity gradient pointed across the boundary. For the second property, the symmetry axis should be fairly smooth so that large changes in direction are discouraged. The third property requires that the length of neighboring rib vectors should

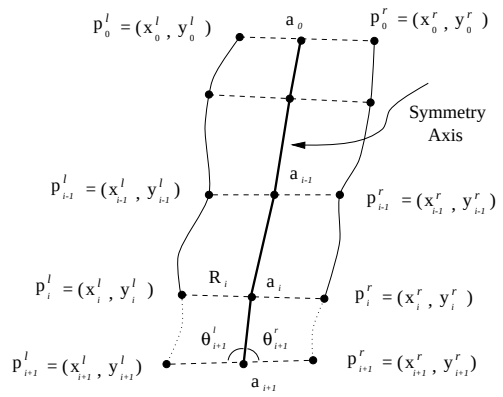

Figure 1. The representation of the shape and the symmetry axis. The left and right boundary points, $\{\vec{p}_i^{\,l}\}$ and $\{\vec{p}_i^{\,r}\}$ are shown. So are the symmetry axis $\{\vec{a}_i\}$ and the rib angles $\{\theta_i^l\}$ and $\{\theta_i^r\}$.

be similar. This will ensure that the boundary of the shape does not fluctuate rapidly. The fourth property states that the vector joining pairs of symmetric boundary points, $\vec{p}_i^{\,l}$ and $\vec{p}_i^{\,r}$, should be roughly perpendicular to the symmetry axis.

After some experimentation, we have settled on the following fitness function

$$E[\{\vec{p}_i^{\,l}\}, \{\vec{p}_i^{\,r}\}] = \sum_{i=1}^{N} E_i(\vec{p}_{i-1}^{\,l}, \vec{p}_{i-1}^{\,r}, \vec{p}_i^{\,l}, \vec{p}_i^{\,r}, \vec{p}_{i+1}^{\,l}, \vec{p}_{i+1}^{\,r})$$

where

$$\begin{aligned} &E_i(\vec{p}_{i-1}^{\,l}, \vec{p}_{i-1}^{\,r}, \vec{p}_i^{\,l}, \vec{p}_i^{\,r}, \vec{p}_{i+1}^{\,l}, \vec{p}_{i+1}^{\,r}) = \\ &\frac{1}{\epsilon + \left|\vec{\nabla} I(\vec{p}_{i+1}^{\,l}) \cdot \widehat{\vec{R}_{i+1}}\right|} + \frac{1}{\epsilon + \left|\vec{\nabla} I(p_{i+1}^{r}) \cdot \widehat{\vec{R}_{i+1}}\right|} \\ &+\lambda_1 |\phi_i(\vec{a}_{i-1}, \vec{a}_i, \vec{a}_{i+1})|^2 + \lambda_2 \left|\left|\vec{R}_{i+1}\right| - \left|\vec{R}_i\right|\right|^2 \\ &+\lambda_3 |\theta_{i+1}^l - \theta_{i+1}^r|^2. \end{aligned} \quad (1)$$

Recall that, in the previous section, the $R_i, \vec{a}_i, \theta_i^l, \theta_i^r$ and ϕ have been defined as functions of the $\vec{p}_i^{\,l}$ and $\vec{p}_i^{\,r}$.

The first two terms of the right hand side of the equation enforces the first property. The third, fourth and fifth terms enforce the second property (smoothness of the symmetry axis), the third property (smoothness of rib vectors), and the fourth property (perpendicularity of axis and rib) respectively.

This goodness of fitness function can be given a probabilistic interpretation using the Gibbs distribution and

Bayes' theorem. In this interpretation, the first two terms of the energy function will correspond to the likelihood function for generating the image. Observe that the terms only constrain the image intensity at the boundaries of the shapes. In the probabilistic interpretation this is equivalent to assuming that the rest of the image is generated by a uniform intensity distribution pointwise. The remaining three terms in the energy function will correspond to the prior probability of the shape.

Symmetry axis and articulation: When applying the symmetric-axis principle to design a segmentation algorithm, one needs to consider the following scenario which is encountered very often. Suppose the image to be segmented contains objects with articulations. Then, we would prefer that a symmetric axis can go along those places where articulations occurred without losing the desired symmetric property. More precisely, in Figure 2, we see, for the symmetry axis to maintain a good correspondences between points on the two boundaries, it is necessary for those points on the outer boundary moving in a faster pace along the corner compared to those on the inner boundary (One can imagine as if there were an inner tangent circle rolling and passing around the curved part of Figure 2). The fitness function in (1) has such property as we have added the perpendicularity term (the last term in (1)) in its definition. Later we will show some experimental results regarding to this aspect.

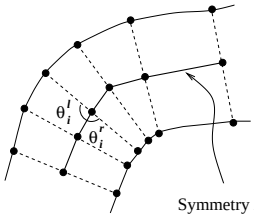

Figure 2. A symmetry axis generated by the left and right boundary moving along corner. This is often encountered for shapes/objects with articulations.

5. Dijkstra's Algorithm with Hashing

We consider the search for the best sequence of pair of points $(\vec{p}_i^{\,l}, \vec{p}_i^{\,r})$, given an initial pair $(\vec{p}_0^{\,l}.\vec{p}_0^{\,r})$. The following graph is constructed. Each node in the graph represents a pair of points $(\vec{p}_i^{\,l}, \vec{p}_i^{\,r})$ from the image plane (see Figure 3). Thus, each node represents a point in a four dimensional space (since a pair of points from *two* dimensional lattice

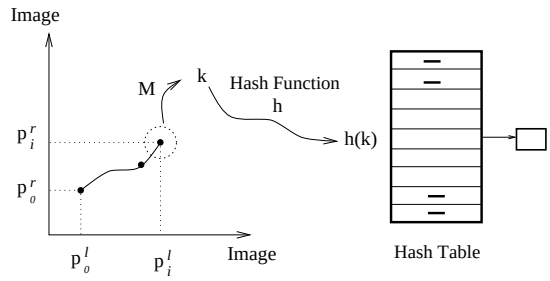

Figure 3. The diagram for the Dijkstra's algorithm with hashing scheme. Notice that in the image plane, the 8-connected neighbor structure is adopted.

have *four* coordinates). Each graph-edge contains the cost computed from the fitness function (1) to select a pair of nodes as part of the segmentation solution.

To segment a region in an image, all possible paths are generated starting from the initial node $(\vec{p}_0^{\,l}.\vec{p}_0^{\,r})$. Each path corresponds to a candidate segmentation. Notice that, without considering occlusion, every path is a continuous curve in a four dimensional space as seen in Figure 3. A segmentation is derived whenever a path has its left and right boundary, thus their symmetry axis, all reaching a same node in the graph. The solution we look for is an optimal path/segmentation with minimal cost.

The Dijkstra's algorithm is implemented with a Fibonacci heap, so it leads to a very efficient optimization computation. To resolve the massive amount of memory required for a global optimization process over a graph with four-dimension nodal structure, the *hashing with chaining* scheme is adopted. Every four dimensional node in the Dijkstra diagram is mapped by a function M to a key, say k, as the input of a well-defined hash function h. In this way, the program is not limited by the size of RAM and can handle images of almost all sizes.

$$(\vec{p}_i^{\,l}, \vec{p}_i^{\,r}) \xrightarrow{M} k \xrightarrow{h} \text{Hash Table}\,.$$

The hash function used in our program is based on the *division method.* An alternative choice is to derive a hash function using the *universal hashing* approach. To construct a good hash function is a pivotal issue, we will not go into details and refer the readers to, e.g., [6].

We have tested our software on both the Silicon Graphics challenge GR and Pentium II PCs. The experimental results are shown in next section.

(a) (b) (c)

Figure 4. Examples of segmentation.

6. Experimental Results

A variety of experiments have been carried out to test the segmentation algorithm. Due to the limited space, we show two sets of experimental results here.

The results in Figure 4 are to illustrate the basic idea of how seeking the symmetry axis can help to segment an image. It can be seen that the algorithm succeeds both at segmentation and at symmetry axis detection and representation. If we relax the straightness (or smoothness) restriction, we derive a segmentation such as the one in Figure 4(b). On the other hand, if the straightness condition is emphasized, the algorithm may override the local edge information in order to find a good symmetry axis. In Figure 4(c) the algorithm creates an artificial "second thumb" on the right hand side so as to be symmetric with the real thumb (on the left).

The second set of experiments is to segment images containing human objects. The test images are Figure 5(a), 5(b) and 5(c). Let us first look at their gray-level gradient images, shown in Figure 5(d), 5(e) and 5(f), respectively (The images are generated from a gradient operator provided in KHOROS 2.1). Though the human object in each image appears to be perceptible, it is difficult to group edges or segment the images correctly. The problem is that , for example, in Figure 5(a) and 5(b), to segment the legs shown in either image, there are other significant edges at junction points stronger than the desired ones. Also, the edges along the legs break where can be seen from the gradient images. With symmetry axis, our algorithm is able to locate and segment symmetric parts of the human shape correctly as displayed in Figure 6.

Figure 6(f) shows that the symmetry axis is detected even when the axis is bent, sometimes quite severely. This supports the observation we make to utilize the perpendicularity property to help the axis moving around the curved parts of an articulated shape.

7. Discussion

This paper describes a method for simultaneously segmenting shapes and detecting their symmetry axes. Our current work involves generalizing this work in three directions. Firstly, the model should be generalized to include regional properties which can be used to help the segmentation. Secondly, the model must be extended to allow for bifurcations in the symmetry axis. Thirdly, we must develop techniques for automatically initializing the algorithm. The first two of these problems can be solved by modifying the goodness of fitness function and increasing the state space. The third problem can be solved by using corner detectors as initial guesses and dynamically selecting the right ones using the procedure described in [10]. Alternatively we could use dynamic programming to search for symmetry axes everywhere in the image by adapting the work of [9] and/or [8].

Acknowledgement

This research was supported by NSF grant IRI-9700446 and AFOSR grant F 49620-96-1-0028

References

[1] M. Barzohar and D. B. Cooper, "Automatic Finding of Main Roads in Aerial Images by Using Geometric-Stochastic Models and Estimation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 459-464, 1993.

[2] H. Blum, "Biological Shape and Visual Science", *J. of Theoretical Biology*, **38**:205-287, 1973.

[3] H. Blum, "A Geometry for Biology", *Annals of the New York Academy of Sciences*, Vol. 231, pp. 19-30, April 1974.

[4] H. Blum and R.N. Nagle, "Shape Description using Weighted Symmetric Axis Features", *Pattern Recognition*, Vol. 10, pp. 167-180, 1978.

[5] M.J. Brady, and H. Asada. Smooth Local Symmetries and Their Implementations. *Int. J. of Robotics Reg.* 3(3). 1984.

[6] T. H. Cormen, C. E. Leiserson and R. L. Rivest, *Introduction to Algorithms*, McGraw-Hill, 1990.

[7] T.J. Cham and R. Cipolla, "Geometric Saliency of Curve Correspondences and Grouping of Symmetric Contours", *Proc. 4th European Conf. on Computer Vision*, Lecture Notes in Computer Science 1064, pp383-398, London, 1996.

[8] J. Coughlan. "Global Optimization of a Deformable Hand Template Using Dynamic Programming." Harvard Robotics Laboratory. Technical report. 95-1. 1995.

[9] D. Geiger, A. Gupta, L.A. Costa, and J. Vlontzos. "Dynamic programming for detecting, tracking and matching elastic contours." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-17, March 1995.

[10] D. Geiger and T. Liu, "Recognizing Articulated Objects with Informational Theoretical Methods", *Proceedings 2nd International Conference on Automatic Face and Gesture Recognition*, pp. 45-50, Killington, Vermont, 1996.

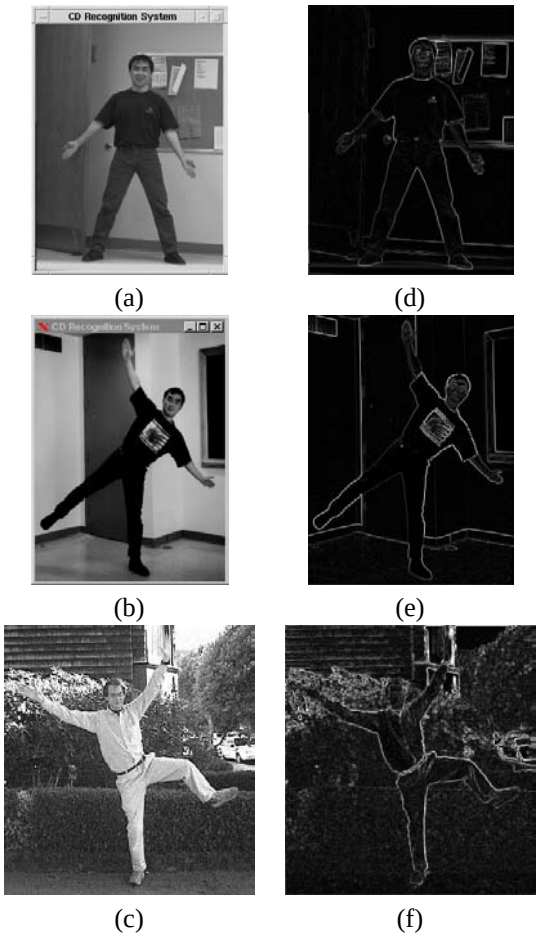

Figure 5. (a), (b) and (c) are testing images. (d), (e) and (f) are their respective (gray-level) gradient images.

[11] M.M. Fleck, D.A. Forsyth, and C. Bregler. "Finding Naked People". *Computer Vision ECCV'96*. Springer Lecture Notes in Computer Science 1065. 1996.

[12] U. Grenander, Y. Chow, and D. Keegan. *HANDS*. Springer-Verlag. New York. 1991.

[13] M. Leyton. Symmetry, Causality, Mind. MIT Press. Cambridge, Mass. 1992.

[14] T. Liu, D. Geiger and R. V. Kohn, "Representation and Self-Similarity of Shapes," *ICCV98*, Bombay, India, January 1998.

[15] U. Montanari. "On the optimal detection of curves in noisy pictures." *Communications of the ACM*, pages 335–345, 1971.

[16] R.L. Ogniewicz, "Discrete Voronoi Skeletons", Hartung-Gorre, 1993.

[17] S.M. Pizer, W.R. Oliver and S.H. Bloomberg, "Hierarchical Shape Description via the Multiresolution Symmetric Axis Transform", *IEEE PAMI*, Vol. 9, No. 4, July 1987.

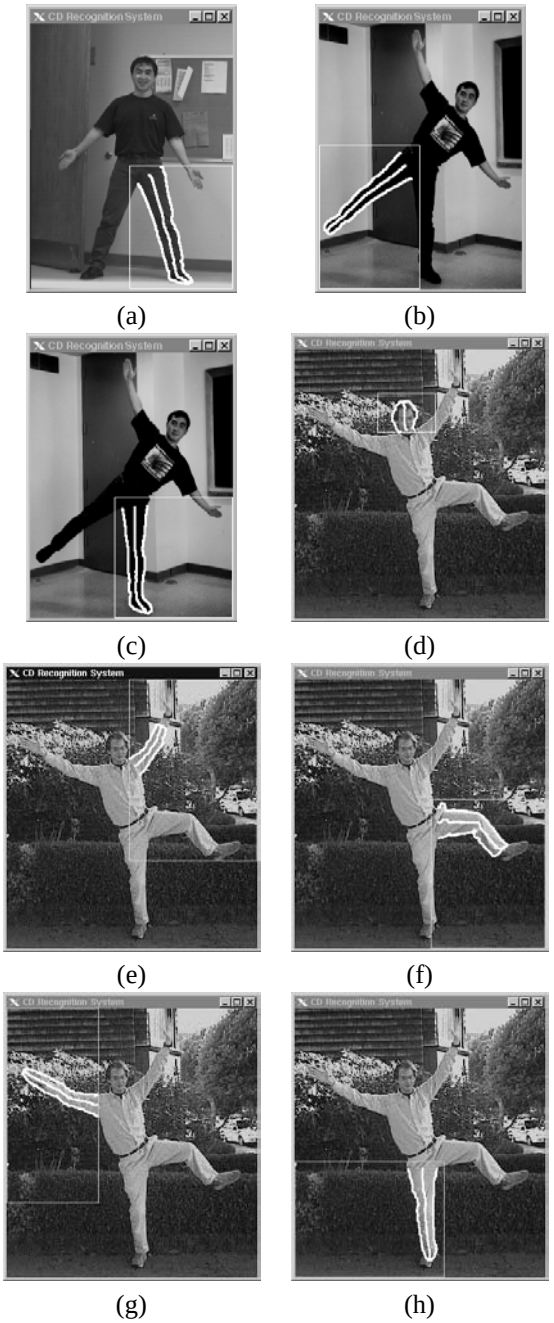

Figure 6. With input images, (a), (b) and (c) in Figure 5, the segmentation results of selected regions are shown in (a)-(h). Compared to the gradient images in Figure 5, we see that segmentation via symmetry axis can avoid the problems caused by bifurcations at the junction points or noise shown in each image.

[18] H. Rom and G. Medioni. Hierarchical Decomposition and Axial Shape Description. *IEEE Trans. PAMI* vol 15. No.10. 1993.

[19] K. Siddiqi and B.B. Kimia, "Parts of Visual Form: Computational Aspects", *IEEE PAMI*, Vol. 17, No. 3, pp. 239-251, March, 1995.

[20] D. Terzopolous, A. Witkin,A. and. M Kass. Symmetry-seeking models and 3D object recovery. *Int. J. Comput. Vision.* 1, 211-221. 1987.

[21] Human Symmetry Perception and its Computational Analysis. Ed. C.W. Tyler. VSP, Utrecht, the Netherlands. 1996.

[22] S.C. Zhu and A.Yuille, "FORMS: a Flexible Object Recognition and Modeling System", *Proceedings 5th International Conference on Computer Vision*, Boston, 1995.