

A Generalized Shape-Axis Model for Planar Shapes

Tyng-Luh Liu

IIS, Academia Sinica, Taipei, Nankang 115, Taiwan

liutyng@iis.sinica.edu.tw

Abstract

We describe a generalized shape-axis (SA) model for representing both open and closed planar curves. The SA model is an effective way to represent shapes by comparing their self-similarities. Given a 2D shape, whether it is closed or open, we use two different parameterizations (one parameterization is oriented clockwise and the other counterclockwise) for the curve. To study the self-similarity, the two parameterizations are matched to each other via a variational framework, where the self-similarity criterion is to be defined depending on the class of shapes and human perception factors. Useful self-similarity criteria include symmetry, parallelism and convexity, and so forth. A match is allowed to have discontinuities, and the optimal match can be computed by a dynamic programming algorithm in $O(N^4)$ time, where N is the size of the shape. We use a grouping process for the shape axis to construct a unique SA-tree, however, when a planar shape is open, it is possible to derive an SA-forest. The generalized SA model provides a compact and informative way for 2D shape representation.

1 Introduction

An effective and compact shape representation system is a critical element for various computer vision application and consequently, the subject has been studied extensively [3, 4, 11, 14, 15, 2, 6, 12, 16, 17, 20, 9, 19, 21, 8].

If shapes are represented with global descriptors, then local deformations can not be described so articulated objects tend to be dissimilar since these deformations may change the global appearance of objects considerably while the entire deformation is concentrated on specific points. Alternatively, if shapes are described by local boundary descriptors [2], then they do not account for region information, object parts, and global properties such as symmetries.

A good representation system of shapes should not be sensitive to small changes of appearance; otherwise similar objects of a same class will be represented very differently.

The shape-axis model [9] is a variational framework for 2D closed shapes/curves (see Figure 1) and is considered to be (i) complete, i.e., all the object information is stored in the representation, (ii) simple and compact, that is, redundancies are removed by capturing symmetries, region information, articulations (local deformations), and dividing the object into “object parts” with as few parts as needed for any level of specified details, (iii) stable, i.e., robust under small variations of the shape (including noise variations and articulations) and (iv) easily computable, i.e., easy to find its representation and to manipulate with it.

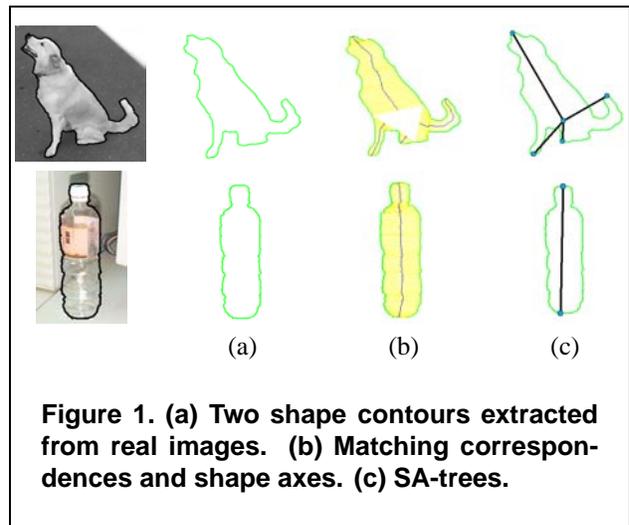


Figure 1. (a) Two shape contours extracted from real images. (b) Matching correspondences and shape axes. (c) SA-trees.

Our goal is to extend the shape-axis model so it can be applied to both *open* and *closed* planar shapes. Compared to other recent related work [17, 19, 20, 21], such extension is significant and is the first shape representation model to be capable of dealing with open shapes. After the shape axis of a shape contour has been recovered, a grouping process based on analyzing the discontinuities of a shape axis can be carried out to construct the unique SA-tree [9]. In case that a planar curve is open it is possible to derive an SA-forest (see Figure 4(c)). With the SA-tree/forest structure, comparing shapes with deformations, articulations and occlusions can be formulated as solving a tree matching process [8].

Previous Work Blum first proposed to describe shapes by their symmetry and thickness [4]. Binford [3] is also a pioneer and brought the attention to generalized cylinders. Other early work includes [1, 11]. Pizer et al. have presented a computational model for object representation via “cores”, or regions of high medialness in intensity images [5]. Leymarie and Levine [10] have simulated the grassfire transform (from the psychology literature). Ogniewicz [12] proposed an efficient Voronoi skeleton algorithm. Siddiqi and Kimia’s work [16] has an interesting mathematical formulation, preserving the intuition of the grassfire idea. It is a development of a reaction-diffusion equation where the symmetry axis is obtained and described by the development of shocks. It is motivated by the framework for shape analysis via shock-diffusion equations [7]. Tari and Shah [19] extended their level-set approach to shapes of arbitrary dimensions. The FORMS by Zhu and Yuille [20] was based on transforming shapes into skeleton graphs for recognition. Recently, Zhu has proposed a stochastic jump-diffusion process for computing medial axes [21]. Liu et al. [9] have addressed the shape representation problem via a variational approach, and it goes beyond the symmetry axis representation. The resulting shape axis can be partially inside and partially outside a shape, and it has a unique SA-tree structure which can be used for shape recognition [8].

2 Variational Matching of Planar Curves

Given a planar curve, two parameterizations, where one is oriented counterclockwise and the other clockwise, can be used to describe it (see Figure 2(a)). Let’s denote them as $\Gamma = \{x(s) : 0 \leq s \leq 1\}$ and $\tilde{\Gamma} = \{\tilde{x}(t) : 0 \leq t \leq 1\}$. We may think of $\tilde{\Gamma}$ as the mirror image of Γ since $\tilde{x}(t) = x(1-t)$, and when the curve is closed we have $x(0) = x(1)$.

By a *match* between Γ and $\tilde{\Gamma}$ we mean a monotone correspondence between the two parameterizations. To represent a match we specify a *pair* of monotone functions $s(\sigma)$ and $t(\sigma)$, each defined for $0 \leq \sigma \leq 1$, such that $x(s(\sigma))$ corresponds with $\tilde{x}(t(\sigma))$. Notice that the end points are not required to be mapped to end points in the other parameterization. It is convenient to represent a match between Γ and $\tilde{\Gamma}$ as a binary function $\mu(s, t)$:

$$\mu(s, t) = \begin{cases} 1, & \text{if } x(s) \text{ corresponds to } \tilde{x}(t) = x(1-t), \\ 0, & \text{otherwise.} \end{cases}$$

The plot of μ is symmetric with respect to the mirror line $s+t=1$ since $\mu(s, t) = \mu(1-t, 1-s)$. Therefore, it suffices to consider only the lower triangular part of the $(s \times t)$ matching space (see Figure 2(b)). The shape axis is defined to be the loci of the midpoints of the correspondences, i.e.,

$$\left\{ \frac{x(s) + \tilde{x}(t)}{2} \mid \mu(s, t) = 1 \right\}.$$

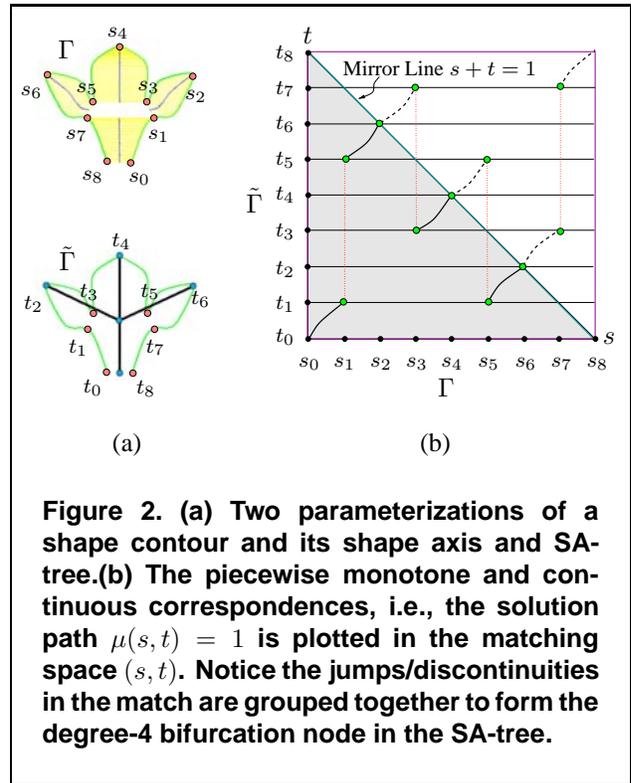


Figure 2. (a) Two parameterizations of a shape contour and its shape axis and SA-tree.(b) The piecewise monotone and continuous correspondences, i.e., the solution path $\mu(s, t) = 1$ is plotted in the matching space (s, t) . Notice the jumps/discontinuities in the match are grouped together to form the degree-4 bifurcation node in the SA-tree.

To determine a “good match” we minimize an appropriate variational problem with the matching energy of the following form

$$\int_0^1 F(x, \tau; \tilde{x}, \tilde{\tau}; s', t') d\sigma, \quad (1)$$

where F is the energy density, and $\tau = x_s/|x_s|$ and $\tilde{\tau} = \tilde{x}_t/|\tilde{x}_t|$ are the oriented unit tangent vectors to Γ and $\tilde{\Gamma}$ at $x(s(\sigma))$ and $\tilde{x}(t(\sigma))$, respectively. Following [9], we define the energy/cost of a match as

$$\begin{aligned} & E[t(\sigma), s(\sigma)] \\ &= \int_0^1 \left\{ F(x, \tau; \tilde{x}, \tilde{\tau}; s', t') + \theta(\sigma) \cdot \text{JumpCost} \right\} d\sigma \\ &= \int_0^1 \left\{ \frac{[(x(s) - \tilde{x}(t)) \cdot (\tau(s)s'(\sigma) + \tilde{\tau}(t)t'(\sigma))]^2}{s'(\sigma) + t'(\sigma)} \right. \\ &\quad + \frac{[(x(s) - \tilde{x}(t)) \cdot (\tau(s)s'(\sigma) - \tilde{\tau}(t)t'(\sigma))]^2}{s'(\sigma) + t'(\sigma)} \\ &\quad + c|x(s) - \tilde{x}(t)|^2 \frac{|s'(\sigma) - t'(\sigma)|^2}{s'(\sigma) + t'(\sigma)} \\ &\quad \left. + \theta(\sigma) \cdot \text{JumpCost} \right\} d\sigma, \end{aligned} \quad (2)$$

where the penalty term $\theta(\sigma) \cdot \text{JumpCost}$ is interpreted as follows: at a jump, the values $t(\sigma+\delta) = \lim_{\delta \rightarrow 0} t(\sigma + \delta)$

and $t(\sigma-) = \lim_{\delta \rightarrow 0} t(\sigma - \delta)$ are different. The associated jump cost can be a function of $|t(\sigma+) - t(\sigma-)|$. In practice we have used a constant cost, `JumpCost`, for jumps. The energy density F defined in (2) is to favor *symmetry* and *closeness* so two points are more likely to match to each other if the distance between them is not large and the corresponding tangents are symmetric. It can be shown that F has several nice structural and geometric properties including parameterization invariant under change-of-variable in σ , translation invariant and rotation invariant.

3 Generalized Shape-Axis Algorithm

We describe a generalized shape-axis algorithm based on *dynamic programming* for solving the variational matching problem (2) for both open and closed planar curves. The method does not put any restriction on the degree of a bifurcation in a shape axis so it can be applied to a variety of complex planar shapes.

In practice a planar curve is given discretely, as a sequence of $N + 1$ points (if it is an open curve of N points, we just copy the first point and append it to the end, that is, $x(s_N) = x(s_0)$). Then it is straightforward to derive a discrete matching energy for equation (2)

$$\begin{aligned}
 & E[t(\sigma), s(\sigma)] \\
 & \approx \frac{1}{N} \sum_{k=0}^{N-1} (1 - \theta(\sigma_k)) \cdot \hat{F}(s(\sigma_k), t(\sigma_k), s(\sigma_{k+1}), t(\sigma_{k+1})) \\
 & + \sum_{\text{jumps}} \text{JumpCost}, \tag{3}
 \end{aligned}$$

where

$$\begin{aligned}
 & \hat{F}(s(\sigma_k), t(\sigma_k), s(\sigma_{k+1}), t(\sigma_{k+1})) \\
 & = \frac{[(x(s(\sigma_k)) - \tilde{x}(t(\sigma_k))) \cdot (\tau(s(\sigma_k))s'(\sigma_k) + \tilde{\tau}(t(\sigma_k))t'(\sigma_k))]^2}{s'(\sigma_k) + t'(\sigma_k)} \\
 & + \frac{[(x(s(\sigma_k)) - \tilde{x}(t(\sigma_k))) \cdot (\tau(s(\sigma_k))s'(\sigma_k) - \tilde{\tau}(t(\sigma_k))t'(\sigma_k))]^2}{s'(\sigma_k) + t'(\sigma_k)} \\
 & + c|x(s(\sigma_k)) - \tilde{x}(t(\sigma_k))|^2 \frac{|s'(\sigma_k) - t'(\sigma_k)|^2}{s'(\sigma_k) + t'(\sigma_k)}, \tag{4}
 \end{aligned}$$

$s'(\sigma_k) = \frac{s(\sigma_{k+1}) - s(\sigma_k)}{N}$ and $t'(\sigma_k) = \frac{t(\sigma_{k+1}) - t(\sigma_k)}{N}$. The indicator function $\theta(\sigma_k)$ is defined to be 1 at σ_k if a jump occurred and 0 otherwise.

Our task now is to search in $\{s_i \times t_j; i, j = 0, 1, \dots, N\}$ for an optimal solution path $\{\mu_{ij}^* = \mu^*(s_i, t_j) = 1 \mid i, j = 0, \dots, N\}$ that yields the minimum energy of (3).

3.1 Dynamic Programming

As stated before, due to the mirror property of μ we shall focus only on the lower triangular part of the diagram $\{s_i \times t_j\}$. Any possible match, including the optimal

one, can be obtained by searching and scanning downward along diagonal lines, 45 degrees, starting from the mirror line $t = 1 - s$ (or $j = N - i$ for the discrete form). Each diagonal line can be indexed by l that varies over $\{0, 1, \dots, N\}$ and represented as $t = \frac{l}{N} - s$. So, $l = N$ corresponds to the mirror line and $l = 0$ corresponds to the origin (s_0, t_0) (see Figure 3). For convenience, we switch from the representation of matching points $(i, j) \rightarrow (s_i, t_j)$ (or $\mu_{ij} = 1$) to the representation $[i, l] \rightarrow (s_i, \frac{l}{N} - s_i)$ (or $\mu_{il} = 1$).

Dynamic programming sets an initial cost along the mirror line $l = N$ and iterates on $l = N - 2, \dots, 0$ (note that $l = N - 1$ does not need to be considered since it will cause a path to reach the mirror line at a non-lattice point, i.e., we can simply set the costs to infinity at the line $l = N - 1$). Along each diagonal line l it visits every $(s_i, \frac{l}{N} - s_i)$, $i = 0, \dots, l$, solving the subproblem ‘‘What is the cost of the best path passing through $(s_i, \frac{l}{N} - s_i)$?’’.

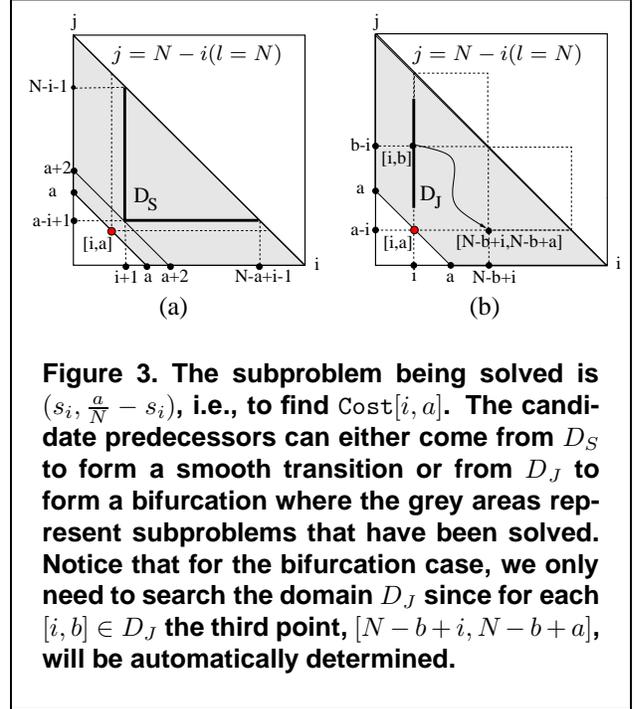


Figure 3. The subproblem being solved is $(s_i, \frac{a}{N} - s_i)$, i.e., to find $\text{Cost}[i, a]$. The candidate predecessors can either come from D_S to form a smooth transition or from D_J to form a bifurcation where the grey areas represent subproblems that have been solved. Notice that for the bifurcation case, we only need to search the domain D_J since for each $[i, b] \in D_J$ the third point, $[N - b + i, N - b + a]$, will be automatically determined.

We use the matrix $\text{Cost}[i, l]$ to store the best cost of each subproblem at $(s_i, \frac{l}{N} - s_i)$. With this representation, the second index gives the index l of the diagonal line and the first index fixes the position on the diagonal line l . A sketch of the algorithm is as follows.

1. Initialization: Initially, all $\text{Cost}[i, l]$'s are set to ∞ except those on the mirror line. The mirror line represents ‘‘self matches’’, i.e., matches between points along the shape contour with themselves. Most often they occur at points of locally maximum/minimum curvature. Thus, one could instead of setting

$$\text{Cost}[i, N] = 0, \quad \text{for } i = 0, \dots, N,$$

put a bias for points along the contour that their curvatures are local maxima/minima, that is, raise the costs for the other points.

2. Iterations on l : Suppose we have solved the subproblems up to line $l = a + 1$, i.e., $\text{Cost}[i, l]$, $i = 0, 1, \dots, l$ and $l = N, N - 1, \dots, a + 1$, are computed. Then, $\text{Cost}[i, a]$, $i = 0, \dots, a$ of diagonal line $l = a$ can be derived by

$$\text{Cost}[i, a] = \min \left\{ G_S[i, a], G_J[i, a] \right\}, \quad (5)$$

where

$$\begin{aligned} & G_S[i, a] \\ = & \min_{[j, b] \in D_S[i, a]} \left\{ \hat{F}(s_i, t_{a-i}, s_j, t_{b-j}) + \text{Cost}[j, b] \right\}, \quad (6) \\ & G_J[i, a] \\ = & \min_{[i, b] \in D_J[i, a]} \left\{ \text{Cost}[i, b] + \text{Cost}[N - b + i, N - b + a] \right. \\ & \quad \left. + [(3 - 2 \times J[i, b] - 2 \times J[N - b + i, N - b + a]) \right. \\ & \quad \left. \times \text{JumpCost}] \right\}, \quad (7) \end{aligned}$$

and

$$\begin{aligned} J[i, a] &= \begin{cases} 1, & \text{if } \text{Cost}[i, a] = G_J[i, a], \\ 0, & \text{if } \text{Cost}[i, a] = G_S[i, a], \end{cases} \\ D_S[i, a] &= \{[i + 1, d] : a + 1 < d \leq N\} \\ & \quad \cup \{[i + k, a + 1 + k] : 0 < k < N - a\}, \\ D_J[i, a] &= \{[i, d] : a + 1 < d < N - 1\}. \end{aligned}$$

The notation G_J represents the cost of grouping the two branches to be part of a bifurcation (due to jumps) and G_S represents the cost of having the predecessor in a smooth path (continuation). D_S and D_J are the domains where dynamic programming searches for possible predecessors to form a smooth path or a bifurcation, respectively (see Figure 3). The binary matrix $J[i, a]$ defined above is to guarantee that each (vertical) jump is penalized only once, and the penalty term in (7) makes sure a degree n bifurcation to be charged JumpCost exactly n times.

To backtrack the optimal solution, the algorithm also keeps the predecessors

$$\text{Back}[i, a] = \begin{cases} \{[j^*, b^*]\}, & \text{if } \text{Cost}[i, a] = G_S[i, a], \\ \{[i, b^*], [N - b^* + i, N - b^* + a]\}, \\ \text{if } \text{Cost}[i, a] = G_J[i, a]. \end{cases} \quad (8)$$

3. Optimal solution and backtracking: In the end this iterative process, from $l = N$ to $l = 0$, will provide $\text{Cost}[0, 0]$, the total cost of the optimal match. The final stage, $l = 0$, requires special attention when the contour is closed and $[0, 0]$ has two ‘‘continuous’’ predecessors, say $J[0, b] = J[N - b, N - b] = 0$. Then unlike the usual ‘‘jump’’ case, it does not imply a bifurcation, due to the periodicity of a closed curve, and in this case the origin does not represent any match, but only the end of the algorithm.

To construct the corresponding SA-tree, we start at the origin $[0, 0]$, and the value $\text{Back}[i, a]$ will take at each step to the predecessor(s). It is either only one predecessor or two, depending on if it is a continuation or a bifurcation. When there is one predecessor it continues accumulating (retrieving) the segment path. When it gives two predecessors we can group them together and argue it has reached a bifurcation node of the tree. To complete a bifurcation node, backtracking should be applied to each of the two predecessors in parallel, and each process is again carried out recursively until reaching a point with only one predecessor. Then all the points visited will be grouped together to complete the bifurcation node, and the backtracking process starting again from each of them.

Complexity: There are $O(N^2)$ subproblems to be solved (evaluating $\text{Cost}[i, l]$ for $i = 0, \dots, l$ and $l = N - 1, \dots, 0$), and the complexity of each subproblem is $O(N^2)$ (the size of total searching spaces D_S and D_J is $O(N)$ and the complexity of $\hat{F}(s_i, t_{a-i}, s_k, t_{d-k})$ is also $O(N)$, assuming uniform matching). Therefore, the complexity of this algorithm is $O(N^4)$.

4 Experimental Results and Conclusion

We have proposed a generalized shape-axis model based on dynamic programming for planar shape representation. The approach generalizes the variational framework presented in [9], and can be used for both open and closed planar shape contours. The shape axis obtained can be interpreted as the optimal solution to a well-defined variational matching problem. This is also where the power of the approach relies, where by changing the optimality criterion we can obtain different representations. Worth noticing that (i) the number of branches in a bifurcation is allowed to be arbitrary, thus data driven and (ii) the shape-axis solution is rather stable against small perturbations on the boundary of a planar shape. A set of experimental results are provided in Figure 4.

Acknowledgments

The author is supported in part by Academia Sinica of Taiwan, and NSC grant 89-2213-E-001-023.

References

- [1] H. Asada and M. Brady, "The Curvature Primal Sketch," *IEEE TPAMI*, **8**(1):2-14, 1986.
- [2] R. Basri, L. Costa, D. Geiger and D. Jacobs, "Determining the Similarity of Deformable Shapes," *Physics Based Modeling Workshop in Computer Vision*, 1995.
- [3] T. O. Binford, "Visual Perception by Computer," *IEEE Conference on Systems and Control*, December 1971.
- [4] H. Blum, "Biological Shape and Visual Science," *J. of Theoretical Biology*, **38**, pp. 205-287, 1973.
- [5] C. A. Burbeck and S. M. Pizer, "Object Representation by Cores: Identifying and Representing Primitive Spatial Regions," *Vision Research*, Vol. 35, 1995.
- [6] Y. Gdalyahu and D. Weinshall, "Measures for Silhouettes Resemblance and Representative Silhouettes of Curved Objects," *ECCV(II)*, pp. 363-375, Cambridge, UK, April 1996.
- [7] B. Kimia, A. Tannenbaum, S. Zucker, "Shapes, Shocks, and Deformations I: The Components of Two-Dimensional Shape and the Reaction-Diffusion Space," *IJCV*, **15**(3):189-224, July 1995.
- [8] T-L. Liu and D. Geiger, "Approximate Tree Matching and Shape Similarity," *ICCV*, pp.456-462, Kerkyra, Greece, September 1999.
- [9] T-L. Liu, D. Geiger and R. V. Kohn, "Representation and Self-Similarity of Shapes," *ICCV*, pp. 1129-1135, Bombay, India, 1998.
- [10] F. Leymarie and M. D. Levine, "Simulating the Grassfire Transform Using an Active Contour Model." *IEEE TPAMI*, **14**(1):56-75, 1992.
- [11] R. Nevatia and T. O. Binford, "Description and Recognition of Curved Objects," *Artificial Intelligence*, Vol. 8, pp. 77-98, 1977.
- [12] R. L. Ogniewicz, *Discrete Voronoi Skeletons*, Hartung-Gorre, 1993.
- [13] A. Pentland and S. Sclaroff, "Closed-Form Solutions for Physically Based Shape Modeling and Recognition," *IEEE TPAMI*, **13**(7):715-729, 1991.
- [14] J. Ponce, and O. D. Faugeras, "An Object Centered Hierarchical Representation for 3D Objects: The Prism Tree," *CVGIP*, **38**(1):1-28, April 1987.
- [15] S. M. Pizer, W. R. Oliver and S. H. Bloomberg, "Hierarchical Shape Description via the Multiresolution Symmetric Axis Transform," *IEEE TPAMI*, **9**(4):505-511, 1987.
- [16] K. Siddiqi and B. B. Kimia, "Parts of Visual Form: Computational Aspects," *IEEE TPAMI*, **17**(3):239-251, 1995.
- [17] K. Siddiqi and B. B. Kimia, "A Shock Grammar for Recognition," *CVPR*, pp. 507-513, San Francisco, CA, 1996.
- [18] D. Terzopolous, A. Witkin, and M. Kass, "Symmetry-Seeking Models and 3D Object Recovery," *IJCV*, **1**(3):211-221, October 1987.
- [19] S. Tari and J. Shah, "Local Symmetries of Shapes in Arbitrary Dimension," *ICCV*, pp. 1123-1128, Bombay, India, 1998.
- [20] S. C. Zhu and A. Yuille, "FORMS: A Flexible Object Recognition and Modeling System," *ICCV*, pp. 465-472, Cambridge, MA, 1995.
- [21] S. C. Zhu, "Stochastic Computation of Medial Axis in the Markov Random Fields," *CVPR*, Santa Barbara, CA, 1998.

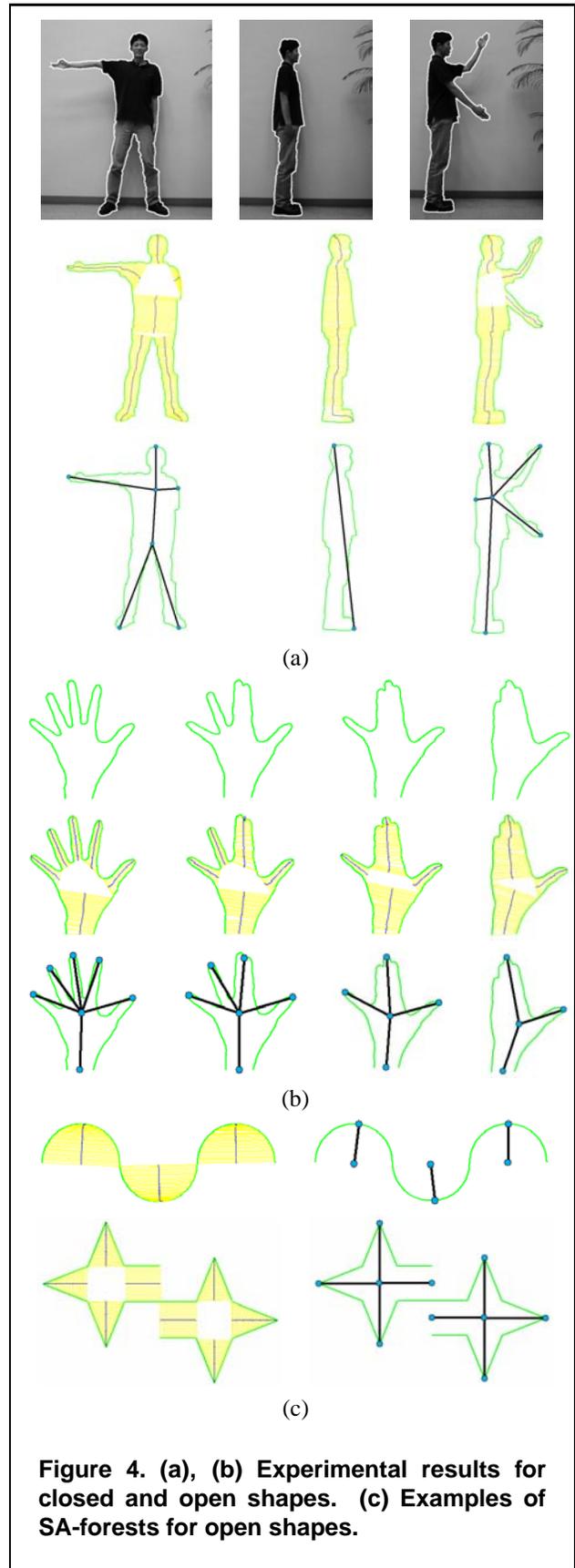


Figure 4. (a), (b) Experimental results for closed and open shapes. (c) Examples of SA-forests for open shapes.