Non-Rigid Body Motion

Andreas R. Dill
Martin D. Levine
Computer Vision and Robotics Laboratory
Department of Electrical Engineering
McGill University
Montréal, Québec, Canada

Abstract

During the last few years motion analysis has developed into a major field of interest in image analysis and understanding. Considerable progress has been made as evidenced by the number of publications. However, most of these have been confined to the study of rigid body motion. Very few have addressed the issue of non-rigid body motion which poses many interesting and difficult problems. We propose a new technique for analyzing non-rigid body motion of closed boundary shapes that is based on the computation of skeletons at multiple resolutions. This technique is used for analyzing the structural changes in the morphology of locomoting lymphocytes and, in particular, of their pseudopods.

Résumé

Depuis quelques années, l'étude de la théorie de l'analyse du mouvement est devenue un domaine d'intérêt majeur en compréhension et analyse d'images. D'immenses progrès ont été réalisés en ce domaine. Ceux-ci sont d'ailleurs mis en évidence par la quantité impressionante de publications écrite jusqu'à ce jour. Cependant la plupart de ces publications ont concerné l'étude des corps rigides. Très peu ont adressé l'étude du mouvement des corps non-rigides, laquelle soulève plusieurs problèmes intéressants et difficiles. Nous proposons une nouvelle technique pour l'analyse du mouvement des corps non-rigides représantables par des contours fermés. Elle est basée sur l'évaluation des skelettes à résolutions multiples. Cette technique est utilisée pour analyser les changements structuraux dans la morphologie des lymphocytes et plus particulièrement de leurs pseudopodes.

KEYWORDS: motion analysis, non-rigid body, skeleton, multiple resolutions, matching

1. Introduction

Methods for motion analysis are quite commonly categorized into two main classes, *intensity based schemes* and *token matching schemes* [Hildreth and Ullman 1982]. The former are based on the computation of local changes in light intensity values whereas *token matching schemes* compare certain previously computed features over time. Any identifiable feature can be chosen as a token no matter how complex it is; that is, the token can be composed of several more primitive features. Tokens that suggest themselves are: termination points of lines and edges, edge segments, boundary segments, points of curvature discontinuity, regions, to name only a few. The main difficulty of this scheme is the correspondence problem. In other words a token must be found at time t_j that correponds to the given token at time t_i $(j > i)$.

Whereas in the case of rigid body motion the tokens between which correspondence must be established are well defined, the choice of these tokens for non-rigid body motion is a very difficult task. For example, to define the tokens in the case of a locomoting blood cell, its shape must be partitioned into meaningful subparts. In general, this problem can certainly not be solved. Part of the difficulty resides in the fact that the notion of *meaningful* depends on the scene. In other words, not only syntactic knowledge is required for partitioning a planar curve but also, to a great extent, semantic knowledge. Thus, Fischler and Bolles [1983] point out that the partitioning problem is *not* a generic task independent of purpose. They also mention that the segments into which a curve is partitioned change when the purpose of the partitioning is altered.

Pseudopods are temporary protrusions of the protoplasm of a cell. Therefore one is interested in partitioning the cell shape into tokens that represent convex subparts of the cell. A typical cell input image is shown in Figure 1.

The symmetric axis transform, introduced by Blum and Nagel [1978], or its discrete version that is usually referred to as skeleton, is a natural way of representing convex subparts of biological shapes. Any locally convex segment of contour, as well as a more globally convex segment of contour that comprises several local convexities, could be a pseudopod. Hence the skeleton would be

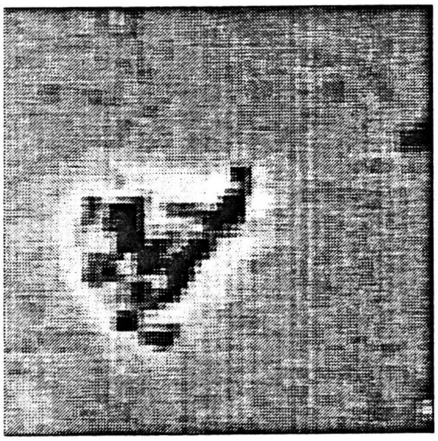

Figure 1 Typical frame of input image sequence

better computed at different resolutions. Since the local as well as the global context must be maintained over time, smoothing operations as proposed by Dyer and Ho [1984] cannot be applied. In their approach small local convexities cause a new skeleton branch only if they are not incorporated in a large global convexity. This is because their approach is based completely on syntactic knowledge.

By computing skeletons at different resolutions a filtered version can be produced without violating the constraints imposed by the semantic knowledge. The resolution at which the shape is examined is related to the degree of smoothing, in that the lower the resolution, the higher the degree of smoothing. Skeleton branches that persist over several resolutions arise from convexities that are locally as well as globally significant. Their stability is related to their perceptual significance. Witkin [1983] proposes a similar stability criterion that depends on the persistence of events over scale changes. In contrast to the shape centered descriptions of which the skeleton is an example, he uses a boundary-based description.

Having computed the skeleton at different scales, we use those computed at the lower resolutions as a measure of how global the underlying convexity is. Clearly the skeletons computed at higher resolutions represent the exact location and orientation of the underlying convexities. The structural changes of the locomoting cell are then quantified by comparing the filtered skeleton version of the cell at different time instances [Dill 1985].

2. Computation of skeletons

The technique used for computing the skeleton is based on an algorithm by Arcelli [1981]. The resulting skeleton can be interpreted as a discrete version of the symmetric axis transform. Given an object as a connected set of pixels O, the skeleton is derived by iteratively tracing its 8-connected contour C. After every iteration, contour pixels that are neither multiple nor lie on a significant convexity are assigned to a set of pixels R, according to some measure of significance that will be explained later. A pixel is multiple if it is traversed more than once during contour tracing, if it has no neighbors in the interior of O (interior: set of pixels $O - C$), or if it has at least one D-neighbor (D-neighbor: horizontally or vertically displaced neighbor of a pixel) which belongs to the contour but is not one of the two direct neighbors along the contour. Before the next iteration, O is assigned to $O = O - R$. The algorithm stops when $R = \emptyset$. The skeleton S then corresponds to the set O. The different sets are illustrated in Figure 2.

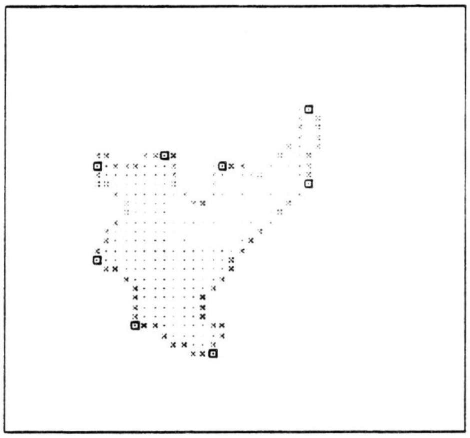

Figure 2 Iteration 1 of the skeletonization algorithm. The contour (set C) is composed of the symbols ⊡ and ×. The set O corresponds to the union of the three different sets of symbols. Contour pixels that lie on a significant convexity are marked with the symbol ⊡. Contour pixels marked as × are assigned to the set R.

The significance of a convexity is measured as a function of the discrete curvature computed at a given contour pixel. At every contour pixel that is not already labelled multiple, i.e., pixels that are assigned to the set R, discrete curvature values are computed. Assuming that the chain code is available, a curvature value is readily obtained if only the two direct contour neighbors, i.e., the preceding and succeeding contour pixel, are considered. The resulting curvature is referred to as the 1-code c_i^1 at contour pixel i. A negative 1-code results from anticlockwise rotation of the contour: -1 correponds to a rotation of -45 deg, -2 to -90 deg, and 3 to 180 deg Positive values range from 0 for a straight line to 4 for an inversion. However, the 1-code cannot be used for evaluating the significance of the underlying convexity because it is too noise sensitive.

The discrete curvature function represented by the 1-code must undergo some smoothing operations in order to be useful. Several methods have been proposed to obtain a more reliable measure of discrete curvature and thereby a better estimate of the significance of the underlying convexity. Rosenfeld and Johnston [1973] introduced k-curvature for this purpose whereas Freeman and Davis [1977] used a similar measure called the incremental curvature. We are smoothing the sequence of 1-codes that represent the discrete curvature along the contour by correlating it with a triangular mask $f_{\Delta_n}(i)$. Its Fourier transform $F_{\Delta_n}(\xi)$ is represented by a quadratic *sinc* function. In other words, the correlation of the 1-code sequence with a triangular mask is equivalent to a low pass filter operation in the frequency domain. Thus,

$$f_{\Delta_n}(i) = \begin{cases} n - |i| & \text{if } |i| \leq n; \\ 0 & \text{elsewhere.} \end{cases} \quad n = 1, 2, \ldots, k \qquad (1)$$

$$F_{\Delta_n}(\xi) = \frac{\sin^2(2\pi\xi(n-1))}{(\pi\xi)^2} \qquad (2)$$

The correlation of the 1-code sequence with the triangular mask $f_{\Delta_n}(i)$ leads to a new discrete curvature measure that was first used by Gallus and Neurath [1970] and given the name n-code c_i^n, defined as

$$\begin{aligned} c_i^n &= f_{\Delta_n}(i) \times c_i^1 \qquad \text{(where: } \times \overset{\text{def}}{=} \text{correlation)} \\ &= \sum_{k=-n}^{k=n} f_{\Delta_n}(k) c_{i+k}^1 \\ &= n c_i^1 + \sum_{k=1}^{n-1} (n-k)\left(c_{i-k}^1 + c_{i+k}^1\right), \quad n \geq 1. \qquad (3) \end{aligned}$$

This n-code can now be used for computing whether a contour pixel lies on a significant convexity. At a given resolution n, contour pixels whose n-codes exceed a certain threshold θ are considered significantly convex. By increasing n, the size of the neighborhood that contributes to the curvature value of a contour pixel grows larger. This is equivalent to lowering the resolution at which the object is examined. If a continuous string of contour pixels is considered significantly convex only the mid-point of this string is labelled convex. New candidates that are too close to a contour pixel that is either labelled multiple or significantly convex are not included. Pixels are regarded as too close if they lie within n contour pixels on either side of a previously selected skeletal pixel.

The set of pixels S which constitutes the skeleton consists of the following types of pixels: pixels with only one neighbor in S which we will refer to as end-nodes; pixels with two neighbors which are normal skeletal pixels; and pixels with three or more neighbors which we

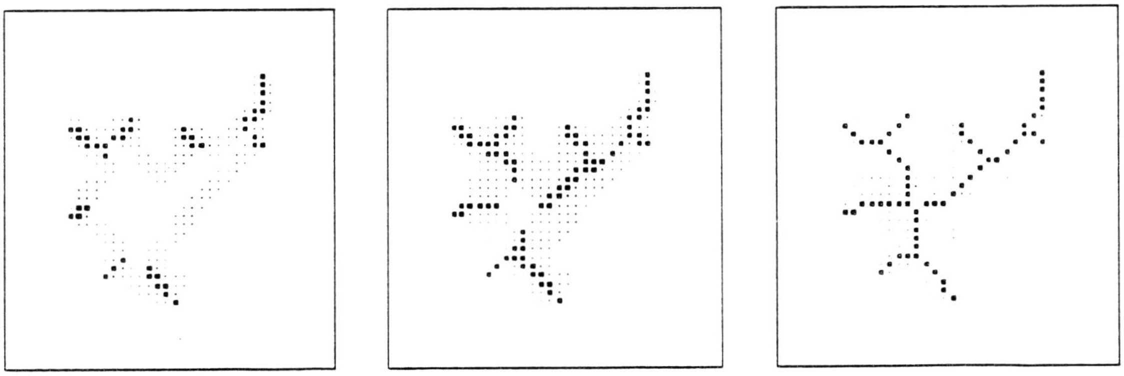

Figure 3 Some iterations of the skeletonization algorithm. The rightmost diagram shows the final skeleton with all end-branches of length 1 deleted. Skeletal pixels are depicted as ■.

shall call branch-nodes. Skeleton branches that emerge from branch-nodes and lead into end-nodes are called end-branches.

Once the set R is empty some postprocessing techniques are applied. First, lines that are of width 2 are thinned without violating the connectivity constraints and then spurious branches of length 1 are deleted. In Figure 3 a few iterations of the algorithm are shown.

3. Representation at multiple resolutions

Before the skeletons computed at different resolutions can be related, another aspect needs some consideration. The question arises whether the numerical values of the n-codes computed at different scales have the same meaning. In other words, will a given contour point on a piece of arc with constant curvature result in the same absolute n-code values for different n's? This should obviously be the case as long as the mask $f_{\Delta_n}(i)$ does not exceed the part of the contour with constant curvature. Quite clearly the definition of the n-code as given in equation (3) violates these constraints. For an arc with constant curvature the resulting n-code values will increase as a function of n. However,it can be shown that this requirement is satisfied by normalizing the triangular mask $f_{\Delta_n}(i)$ so that the area between the mask and the x-axis is always 1, independent of n. As a matter of fact the normalized n-code becomes independent of the resolution n [Dill 1985]. Thus, the employed mask is the normalized n-code $\bar{c}_i^n$, where

$$\bar{c}_i^n = \frac{c_i^n}{n^2} \qquad (4)$$

These normalized n-codes are used for computing the skeletons at different resolutions as detailed in section 2.

In other techniques that work with multiple resolutions for representing perceptually prominent points of a planar curve but use a boundary centered description over a shape centered description, the comparison between different resolutions can be quite tedious [Witkin 1983]. This is because prominent points along a planar curve, whether they correspond to concavities, convexities or to zero crossings of the curvature function, will be displaced with respect to their true physical position as the resolution is lowered. The position of the point that reflects a certain feature is only equivalent to the exact position at a very high resolution.

In contrast to the above, the skeletons computed at different resolutions can be compared in a straightforward manner. It is true that the end-nodes of the end-branches also change position as a function of resolution but the corresponding branch-nodes do not alter as long as the accompanying end-branch persists. Because of this, it is sufficient to compare the coordinates of the branch-nodes from which the end-branches emerge in order to verify whether a convexity persists over a certain range of resolutions.

In the following example (Figure 4), the skeleton is

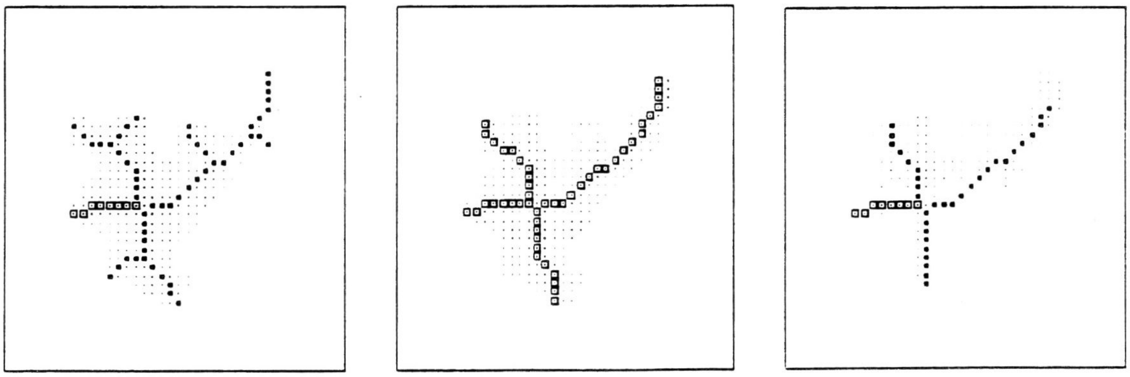

Figure 4 Skeleton filtering process.
The skeleton to the left was computed at $n = 5$ and the skeleton in the middle at $n = 7$. Both were computed with a threshold of $\theta = 0.12$. The skeleton to the right is the filtered version. Branches at the high resolution that match a branch at the low resolution are characterized by the symbol □. Non-matching branches that are deleted or merged at the high resolution and thus newly created in the filtered version are depicted as a sequence of ■.

computed at two different resolutions: $n = 5$ and $n = 7$. At both resolutions the threshold is set to $\theta = 0.12$. Convexities that persist over both scales, hence signalling a global convexity, are included in the filtered version. Local convexities that appear only at the smaller scale are filtered. If two end-branches that are considered local share the same branch-node, they are merged creating a new end-node that lies between the old end-nodes; otherwise, they are deleted. The leftmost diagram of the figure corresponds to the unfiltered version of the skeleton computed at the small scale and the diagram in the middle to the unfiltered version at the large scale. The rightmost diagram of the figure shows the filtered version of the skeleton. The branches displayed with the symbol ▪ in the unfiltered skeleton at the small scale refer to branches that are either deleted or merged. In the filtered version they refer to the corresponding new end-branches. Matching branches that are not altered are displayed as ⊡.

4. Conclusions

We have presented a new technique for representing perceptually salient points on a closed planar curve with skeletons computed at different resolutions. One of the merits of this technique is the ease with which descriptions at different scales can be compared. The representation of convexities by branches of a skeleton is particularly well suited for analyzing non-rigid motion. That is because they do not only correspond to locally and globally significant convexities but also preserve the true orientation (as opposed to the orientation of the convexity with respect to the center of the shape) of these subparts.

Acknowledgements

This research was supported in part by the Medical Research Council of Canada under Grant No. MA-3236, and the Province of Quebec under Grant No. EQ-633. A.R. Dill was in part supported by the Canadian Government Award for foreigners. M.D. Levine would like to thank the Canadian Institute for Advanced Research for their support.

References

Arcelli, C., 1981, *Pattern Thinning by Contour Tracing*, Computer Graphics and Image Processing, Vol. 17, pp. 130-144

Blum, H. and Nagel, R.N., 1978, *Shape Decription Using Weighted Symmetric Axis Features*, Pattern Recognition, Vol. 10, pp. 167-180

Dill, A.R., 1985, *Expert Syst́em for the Analysis of Cell Locomotion*, M.Eng thesis, Computer Vision and Robotics Lab, McGill University, Montréal, Québec, Canada, in preparation

Dyer, C.R. and Ho S., 1984, *Medial-Axis-Based Shape Smoothing*, Proceedings of the 7^{th} ICPR, Montréal, Québec, Canada, pp. 333-335

Fischler, M.A. and Bolles, R.C., 1983, *Perceptual Organization and the Curve Partitioning Problem*, Proceedings of the 8^{th} IJCAI, Karlsruhe, West Germany, pp. 1014-1018

Freeman, H. and Davis, L.S., 1977, *A Corner-Finding Algorithm for Chain-Coded Curves*, IEEE Trans. on Computers, Vol. C-26, No. 3, pp. 297-303

Gallus, G. and Neurath, P.W., 1970, *Improved Computer Chromosome Analysis Incorporating Preprocessing and Boundary Analysis*, Phys. Med. Biol., Vol. 15, pp. 435-445

Hildreth, E.C. and Ullman, S., 1982, *The Measurement of Visual Motion*, A.I. Memo No. 699, MIT Artificial Intelligence Lab, Cambridge, Mass.

Rosenfeld, A. and Johnston, E., 1973, *Angle Detection on Digital Curves*, IEEE Trans. on Computers, Vol. C-22, pp. 875-878

Witkin, A., 1983, *Scale-Space Filtering*, Proceedings of the 8^{th} IJCAI, Karlsruhe, West Germany, pp. 1019-1022