COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING 47, 203–226 (1989)

Shape Matching Using Curvature Processes

EVANGELOS E. MILIOS

Department of Computer Science, University of Toronto, Toronto, Canada M5S 1A4

Received April 27, 1988; accepted December 13, 1988

Shape matching is a fundamental problem of vision in general and interpretation of deforming shapes in particular. The objective of matching in this instance is to recover the deformation and therefore generalizes the notion of correlation, which aims to only produce a numerical measure of the similarity between two shapes. To address shape matching, we introduce a new representation of a closed 2D shape as a cyclic sequence of the extended circular images of the convex and concave segments of its contour. This representation is then used to establish correspondences between segments of the two contours using dynamic programming. Finally, we compute a recovery of the differences between two similar contours in terms of the action of curvature process. Computation of convex and concave segments of the contours, given in piecewise linear form, is accomplished using the analytic representation of a local B-spline fit. We show the result of our deformation recovery scheme appliedto dynamic cloud silhouette analysis using hand-traced input from real satellite images.

1 INTRODUCTION

Planar shape matching is an important problem in computer vision, with broad applications to signal understanding tasks such as character and schematics recognition, as an approach to solving the correspondence problem in sequences of intensity images and in the interpretation of time varying signals, such as X-ray cardioangiograms [21] and dynamic analysis of clouds in satellite images [23, 3]. Pavlidis [14] gave an overview of planar shape analysis until 1980. He classifies relevant work in two broad categories, those operating in a transform domain and those operating in the space domain. The latter are subdivided into external, which follow the contour, and internal or global, which transform pictures into relational graphs or skeletons. Since 1980, both external and internal methods have been pursued. External methods have primarily relied on the concept of scale-space filtering [24, 14], while internal methods have refined the concept of local symmetry axis transforms [4].

Different shape techniques have different shape matching methods naturally associated with them. Shape represented as a relational graph calls for graph matching [19]. Explicit representation of relations between shape features, such as angles, implies matching by relaxation methods [5]. Shape represented by a sequence of features along its contour suggests string matching [14]. Viewing the shape as a function in $\Re^2$ or $\Re^3$ leads to the notion of search for a deformation function that will map the two shapes to be matched onto each other. Searching for a deformation function has been formulated as a variational constraint optimization problem, that may incorporate both similarity and smoothness constraints [24]. An early approach to dynamic shape matching used a piecewise linear approximation to the angle that the tangent forms with a reference direction as a function of arc length [27]. The slope of this function represents curvature, and the matching algorithm in [27] uses the slope and length of the line segments of the approxima-

0734-189X/89 $3.00

tion, and the order, in which they appear. A multiscale shape representation was used in [28], and the associated matching method relied on explicit matching of zero crossing contours in scale space.

Our approach to shape matching follows the contours of planar shapes, therefore it is external, according to the classification of [14]. Our work combines the following aspects: (a) the well-recognized importance of curvature variation along the contour [2, 14], (b) recent work on curvature-based primitives [7], (c) the view of curvature variation as a result of processes acting on the shape [10], (d) the notion of Gaussian image in computer vision as an orientation-preserving representation of convex parts (Ch. 16 of [8]), and (e) the use of higher-order shape models, which allow more reliable estimates of curvature than polygonal approximations [14].

More specifically, we address the problem of matching two smooth closed contours, which are related by deformations that alter their curvature structure. Deformations typically include protrusions, a result of internal resistance, and indentations, a result of squashing. Our approach is to first establish correspondences between segments that have not drastically deformed (in the sense of introducing or eliminating protrusions and indentations) and then to explain how the remaining segments have deformed in terms of productions of a process grammar [10].

A natural question associated with our approach is that of uniqueness of the resulting interpretation, because the process grammar is ambiguous, leading to multiple solutions. Therefore criteria are required for selecting a preferred interpretation. Such criteria must be both computationally feasible and perceptually relevant. The criterion we propose involves metric information extracted from the extended circular images of the segments that take part in a hypothesized production of the process grammar.

The first step in our method is to model shape in terms of the appropriate shape primitives. We use contour segments bounded by perceptually relevant inflection points as our shape primitives. A closed contour is thus described as a circular sequence of alternating convex and concave segments. We assume that such analysis takes place at the proper scale, which is known a priori and has been used to smooth both contours.

The second step is to establish a direct correspondence between segments that have not deformed drastically; i.e., they have preserved their approximate absolute normal or tangent orientation. We use a standard dynamic programming approach to establish non-crossing correspondences. Our distance function depends on the similarity of the extended circular images of two segments. The outcome of this direct correspondence is pairs of associated segments in the two contours and unassociated segments.

The third step is to account for the unassociated segments by viewing them as a result of processes that have acted upon the shapes. The account we are looking for is in the form of productions of a process grammar that explain unassociated segments as results of squashing or internal resistance. To discover such processes we use a greedy algorithm, that alternates between dynamic programming for establishing direct correspondences, and discovery of processes, that alter the two shapes. At each iteration of the algorithm, a direct correspondence is established between the current two shapes, and a single process, selected among all possible processes, is effected, that modifies one of the two shapes. The algorithm stops when

no more processes can be found, i.e., when all unassociated segments have been accounted for, or when none of the possible processes is acceptable.

A fundamental question about the matching algorithm is how many segments need to be matched initially. The algorithm works by discovering processes which replace existing associations with better associations, and thereby improve the quality of matching (defined in terms of the similarity of the extended circular images of pairs of segments). Therefore, the two contours being matched should have a sufficient number of associated segments to allow discovery of processes that improve such associations. Otherwise, the matching algorithm will be unable to find processes that improve associations, and the result will be that most segments will remain unassociated. In this paper, we assume that deformations are local in character, and therefore similarity of absolute orientations of segments is one of the criteria for judging segment similarity. Absolute orientation is conveniently expressed by the domain of definition of the extended circular image.

The paper is structured as follows: Section 2 summarizes the concept of extended circular image of a smooth planar shape, closed or open, and discusses possible discrete approximations. Section 3 relates curvature processes with the extended circular images of the participating segments. Section 4 describes the matching algorithm, which is based on an "associate-hypothesize-test" loop.

2. EXTENDED CIRCULAR IMAGE OF A PLANE CONTOUR

Circular images are the 2-dimensional analog of Gaussian images for 3-dimensional surfaces [8]. Given a plane curve, we map each point of the curve onto a point on the unit circle that has the same normal (or tangent) as the point on the original curve. In the case of a closed convex curve with continuous tangent everywhere and no zero-curvature segments there is a one-to-one mapping between points on the curve and points on the unit circle. Zero curvature or straight line segments map to a single point on the unit circle. A useful property of the circular image is that it rotates with the curve by exactly the same amount. Therefore, the circular image of a convex segment encodes information about the absolute orientation of the segment. The same properties hold if the circular image is defined as a mapping of each point of the curve onto a point on the unit circle, the normal of which is the same as the tangent of the point of the original curve. The latter definition, defined as the tangent circular image in [13], yields a circular image that is rotated by $\pi/2$ with respect to the former definition.

Proceeding in a fashion analogous to that for extended Gaussian images of 3-dimensional surfaces, we can give formal definitions for the extended circular image of a plane contour [8]. The extended circular image (ECI) of a segment of constant curvature sign is defined to be the radius of curvature as a function of tangent orientation, encoded as an arc length on the unit circle. Equivalently, the extended circular image can be defined by assigning a mass to the image of a curve point on the unit circle equal to its radius of curvature, i.e., the inverse of curvature. Thus we have the following definition:

DEFINITION 1. The *extended circular image* of a curve segment of constant curvature sign is defined as the function that maps the tangent orientation of the curve at point S, $\theta(S)$, to the radius of curvature of the curve at the same point S,

$\rho(S)$. Therefore, the following equation holds:

$$\rho(\theta(S)) = \frac{1}{\kappa(S)}. \tag{1}$$

The mapping of $\theta(S)$ to $\kappa(S)$ is the *inverse extended circular image*.

Fundamental properties of the extended circular image can be obtained by integrating over curve length or angle. The curvature κ of a plane curve is defined as the limit of the ratio of the change in tangent orientation to the length of the segment, as the length of the segment approaches zero:

$$\kappa = \lim_{\delta s \to 0} \frac{\delta\theta}{\delta s} = \frac{d\theta}{ds} \tag{2}$$

$\delta\theta$ is equal to the length of the circular image of the infinitesimal segment. If we integrate the above differential relationship over a finite segment S of the curve, we obtain

$$\int_S \kappa \, ds = \int_\Theta d\theta = \Theta, \tag{3}$$

where Θ is the length of the circular image of the curve segment.

If we integrate over the inverse of curvature $1/\kappa$, we obtain

$$\int_\Theta \frac{1}{\kappa} \, d\theta = \int_S ds = S \tag{4}$$

i.e., the length of the finite curve segment.

Plane curves are commonly parameterized by curve length, and they are represented as curvature versus curve length [1]. The fact that points of segments of constant curvature sign can be mapped one-to-one into the unit circle suggests an alternative parameterization of such segments, namely in terms of arc length on the unit circle. The potential advantages of such a parameterization are that it is insensitive to scale changes in the image, or non-uniform stretching or shrinking of the curve, and that it encodes absolute orientation information. Therefore, it may be more suitable for matching deforming shapes than the curve length parameterization most commonly used. The disadvantage is that the mapping into the unit circle is one-to-one only for segments of constant curvature sign.

It can be argued [11] that an extended circular image uniquely specifies a closed convex curve, and computational schemes can be devised for recovering the curve from its extended circular image. As mentioned in [8], the extended circular image of a closed convex figure has some interesting properties. The first is that the center of mass of the extended circular image of a closed figure is at the center of the unit circle. The second is that the total mass of the extended circular image is equal to the total length of the curve (a corollary to a previous derivation). The third is that convolution of the extended circular image of a closed figure with an arbitrary

positive function defined on the unit circle produces a new extended circular image that corresponds to a smoothed version of the original curve.

In this paper we are primarily concerned with extended circular images of curve segments of constant curvature sign. The extended circular image of an open convex curve (defined as a curve, the tangent of which rotates always in the same direction as we move from one endpoint to the other) is defined only on a connected subinterval of the unit circle. In the case of a spiralling convex open curve, the above subinterval may be wider than 2π. Since we are dealing with open curves, it is useful to generalize the concept of rotation index to non-integer values. Before we can do so, we must deal with the problem that the angle $\theta(S)$ between the tangent of a curve at S and the horizontal is not necessarily a continuous function of the position of S, due to a possible phase wrapping by a multiple of 2π. In [13], it is argued that $\theta(S)$ can be adjusted (unwrapped) so that it becomes a continuous function of the curve's independent parameter, such as curve length. The continuous version of tangent orientation $\theta(S)$ is used to define the rotation index of a curve segment.

DEFINITION 2. The *rotation index* of an open curve segment with starting point S and end point E is the number $(\theta(E) - \theta(S))/2\pi$.

The rotation of a closed plane curve is an integer, and according to the rotation index theorem [13], the rotation index of a simple closed plane curve has absolute value equal to 1 (informally, a simple closed plane curve is a non-self-intersecting curve). The domain of definition of the extended circular image of a convex curve segment has length equal to the rotation index of the segment multiplied by 2π.

Based on the previous discussion, we define the following representation for simple closed plane curves.

DEFINITION 3. The *extended circular image representation* of a simple closed plane curve is defined as a cyclic sequence of the extended circular images of its convex and concave segments, obtained by segmenting the curve at its inflection points.

If we are to compute with extended circular images of contour segments, we need to approximate them by discrete functions. This requires sampling of the independent variable of an ECI, namely arc length on the unit circle. As its continuous counterpart, a discrete ECI approximation is required to have an integral $\int_{-\infty}^{\infty}\rho(\theta)\,d\theta$ equal to the length of the curve, where $\rho(\theta)$ is a sum of the discrete values each represented as an impulse.

One possible sampling strategy is obtained by fixing $\Delta\theta$. This approximation is obtained by following the continuous contour and checking off the points at which the tangent has changed direction by $\Delta\theta$ with respect to the previous point. Each contour segment thus obtained is mapped onto an impulse on the unit circle. The discrete ECI that results consists of uniformly spaced impulses (one impulse every $\Delta\theta$), with each impulse having strength equal to the length Δl of the segment of the curve. Highly curved portions of the contour will have low strength impulses, whereas almost straight segments will have high strength impulses.

Another sampling strategy is obtained by fixing Δl. This time we traverse the contour and we check off points at which we have covered contour length equal to Δl with respect to the previous point. The resulting discrete ECI consists of

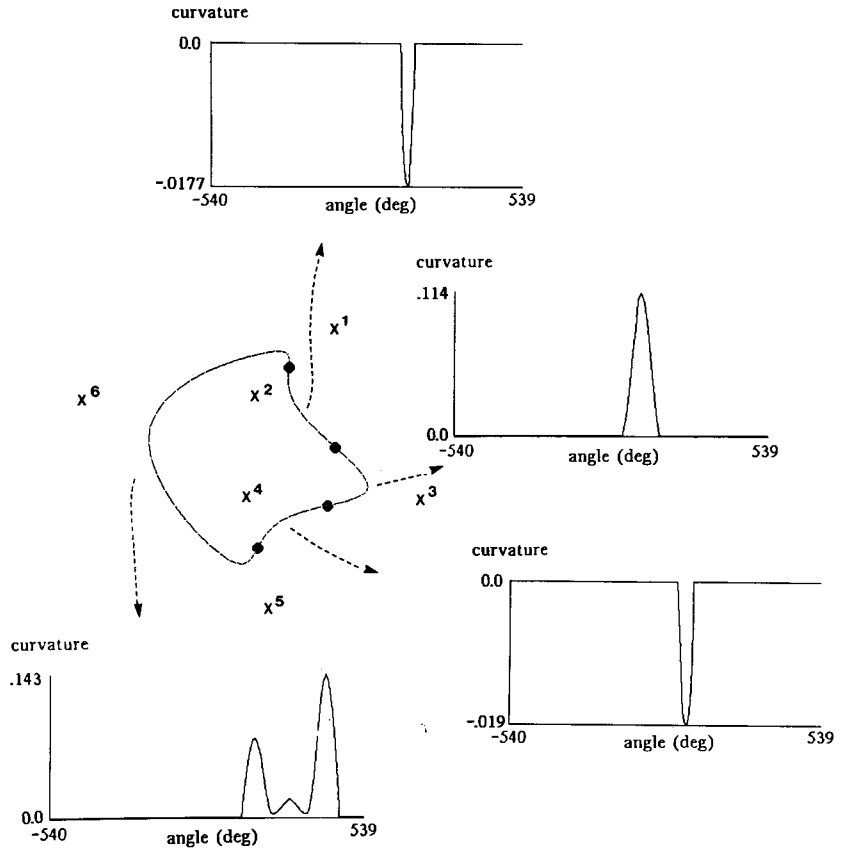

FIG. 1. Convex and concave segments of a closed contour, and the corresponding discrete approximations to their extended circular images. Dots represent inflection points, whereas x's denote knot points. Extended circular images are defined between -540 and 540 degrees. Curvature is measured in radians per unit length, where unit length is the horizontal distance between neighboring pixels.

non-uniformly spaced impulses, all of the same height, equal to Δl. The location of each impulse on the unit circle is the average tangent direction of the corresponding segment. Therefore highly curved portions of the contour give rise to sparsely spaced impulses, whereas in parts of low curvature the impulses are densely spaced.

Both sampling strategies preserve the fundamental property of the ECI of integrating into the length of the original contour. In both cases, the integral is replaced by a summation, since the integrand is a sum of impulses. In the first sampling strategy, we obtain a uniformly sampled digital signal, whereas in the second, the signal is non-uniformly sampled. In the computational implementation of our matching algorithm, we use an approximation to the uniformly sampled ECI. To compute such an approximation, we start with dense, but non-uniform, samples of the ECI obtained from the analytic form of the B-spline representation described in the Appendix. We then compute uniform samples through linear interpolation. Figure 1 shows the segments of a closed piecewise B-spline curve and the associated inverse extended circular images.

3. CURVATURE PROCESSES AND INDUCED EXTENDED CIRCULAR IMAGE OPERATIONS

Leyton [10] introduced the notion of shape as the outcome of processes that formed it and argued that the process history of a shape can be recovered from its curvature extrema. He proposed a process grammar, which describes the effect of processes as modifying the type and/or number of curvature extrema of the shape. The productions of the process grammar are expressed in terms of the following four possible types of curvature extrema, shown together with their semantic interpretation:

- M^+: positive maximum, protrusion.
- m^+: positive minimum.
- m^-: negative minimum, indentation.
- m^+: negative maximum.

Using the above four types of curvature extrema, an elegant process grammar is proposed. Each production of the grammar corresponds to a curvature process. The productions are:

- $Cm^+: m^+ \rightarrow 0m^-0$, squashing continues till it indents.
- $CM^-: M^- \rightarrow 0M^+0$, internal resistance continues until it protrudes.
- $BM^+: M^+ \rightarrow M^+m^+M^+$, a protrusion bifurcates.
- $Bm^-: m^- \rightarrow m^-M^-m^-$, an indentation bifurcates.
- $Bm^+: m^+ \rightarrow m^+M^+m^+$, a protrusion is introduced.
- $BM^-: M^- \rightarrow M^-m^-M^-$, an indentation is introduced.

Our approach in using the concept of shape curvature processes in matching is to devise algorithms that map one shape onto another by applying productions in both directions. It is necessary to consider productions as acting in both directions, since in general there is no guarantee that one of the two shapes being matched is going to correspond to an earlier stage of the evolution of the other shape.

Leyton's process grammar suggests one way of accomplishing this task, namely computing the curvature extrema of the two shapes and then establishing correspondences between them either directly or via productions. On the other hand, Richards *et al.* [17, 7] rely on segments bounded by minima of negative curvature. To describe such segments, they introduce the concept of codons, which are curve segments lying between minima of curvature (not necessarily negative). Codons have zero, one or two curvature zero crossings (inflections) and they yield a total of five codon types, plus the degenerate case of a straight line.

Matching based on curvature extrema is highly sensitive to noise, as it is demonstrated by Fig. 2. In this figure we show two piecewise B-spline curves and the corresponding curvature extrema. We see that very small variations in the curve can result in a different number of curvature extrema and in different positions. Any shape matching method that relies on matching of curvature extrema would have a lot of difficulty accounting for such random perturbations. We would like our primitives to reflect perception in a more robust way, and therefore we opt for segments as our primitives. Further arguments against the use of curvature extrema as contour descriptors are provided in [22].

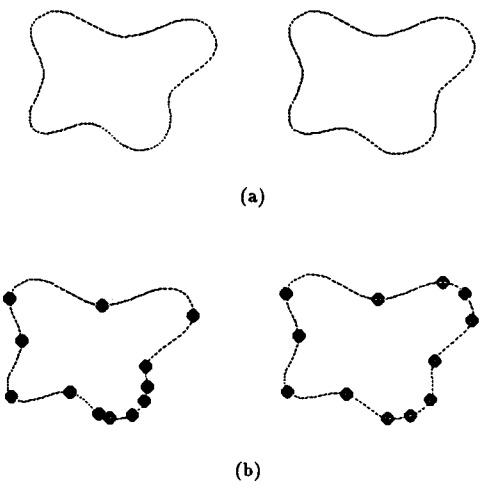

FIG. 2. (a) Two perceptually similar piecewise polynomial contours and (b) their curvature extrema. Both contours consist of local B-splines. Knot points are shown by x's. Note that in spite of the apparent similarity of the two contours, curvature extrema are not necessarily similar in number or location in the two contours.

The problem of using codons as primitives is that codons are too complex to be easy to directly match, because they do not necessarily have constant curvature sign, and therefore the theory of extended circular images is not directly applicable. To combine the simplicity of Leyton's process grammar with the potential stability that segments offer, compared to point primitives, we choose to segment the curve at inflection points, thereby obtaining segments of constant curvature sign. We then compute the extended circular image of each segment and use it for matching.

By adopting a segment-centered viewpoint towards curvature processes, we can give an alternative interpretation of the two types of productions of [10]. Leyton calls the Cm^+ and CM^- productions continuations, and the rest bifurcations. Continuations are processes that continue till they introduce two new inflection points, while bifurcations replace a single curvature extremum by two of the same type. In the context of segments of uniform curvature sign, continuations introduce the splitting of a single segment into two of the same curvature sign separated by a third segment of the opposite curvature sign, as shown in Fig. 3. If we use the symbols C and V to denote convex (positive curvature) and concave (negative curvature), respectively, then we can rewrite the continuation productions of the process grammar as:

$$C \rightarrow CVC \tag{5}$$

$$V \rightarrow VCV. \tag{6}$$

Bifurcations, shown in Fig. 3, on the other hand, do not introduce new inflection points, and therefore they correspond to identity productions in terms of segments, i.e., $C \rightarrow C$ or $V \rightarrow V$. Therefore, the continuation productions stated in terms of segments subsume the corresponding ones in Leyton's process grammar, because

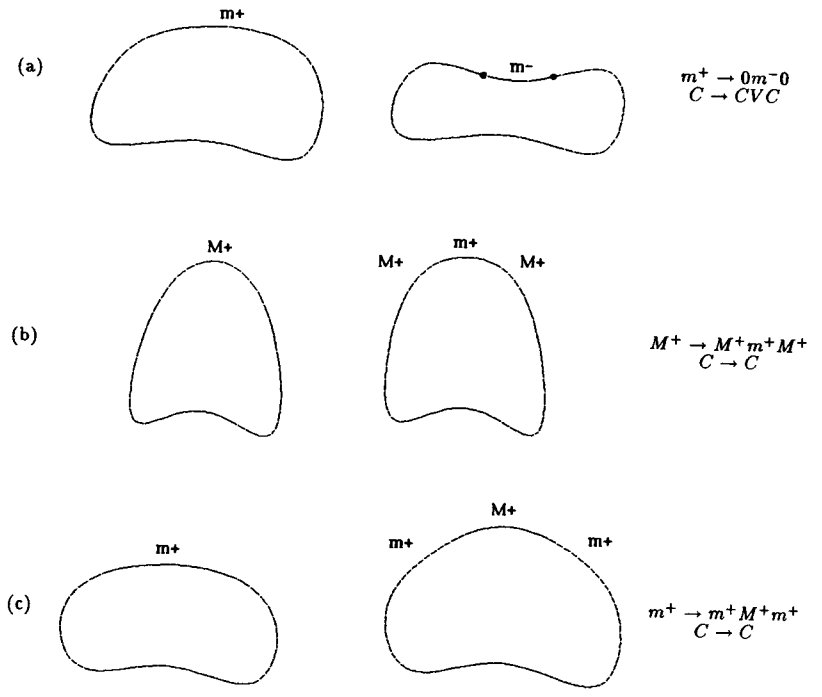

FIG. 3. Examples of curvature processes: (a) continuation, (b), (c) bifurcations. The associated point-based and segment-based productions are shown. Filled dots indicate inflection points. Bifurcations correspond to identity segment-based productions. However, the extended circular images of the two sides of the production have a different number of maxima and minima.

they can also account for an arbitrary number of bifurcations. Viewed this way, continuations and bifurcations belong to two distinct levels of abstraction, suggesting that matching be performed at the continuation level first, and then at the bifurcation level.

Furthermore, we classify bifurcations into two classes. The first class includes BM^+ and Bm^- and corresponds to processes that, if allowed to fully develop, will eventually lead to the introduction of new inflection points. The second class includes Bm^+ and BM^- and it includes processes that will not necessarily lead to new inflection points, if allowed to continue.

Therefore a $C \rightarrow CVC$ production in general includes the effect of one or more BM^+ or Bm^+ productions applied at an early stage of development, followed by a Cm^+ production. Similar observations hold for a $V \rightarrow VCV$ production.

Induced Operations on the Extended Circular Images of the Segments Participating in a Curvature Process

Before we can use the concept of curvature processes for matching shapes, we need a measure of goodness of a production, given the participating segments. A complete physical model accounting for shape deformations should include a model of the material and models for the acceptable forces that induce the deformation. Measuring the goodness of production could then be reduced to comparison of

quantities with physical meaning, such as the energy associated with the deformation. Physical models are available [9], but in their current form the user is required to provide the constraint forces. There is a trade-off between the sophistication of a shape model and the ease of modelling. In this paper, we opt for a simple shape model, *B*-splines, and a crude physical model for the deformation, corresponding to preservation of contour length; i.e., we assume a fixed-length string model for our contours. The fixed-length string model for the deformation allows the definition of a goodness measure, which can be computed using the extended circular images of the participating segments.

In the sequel, we provide a precise definition of the goodness measure of a production. Then we enumerate desired properties of a concrete realization of a curvature process. Finally, we present a simple operation on the ECIs of the participating segments and we discuss the degree to which it satisfies the desired properties.

To place the concept of goodness measure in the proper context, let us assume that we want to assess the goodness of the production $C_1V_1C_2 \rightarrow C_3$, where C_1, V_1, C_2 are consecutive segments of the first contour, and C_3 is a segment of the second contour. The production can be thought of as mapping the extended circular images of C_1, V_1, C_2 onto the extended circular image of C_3. The goodness measure is then defined as the distance between the ECI of C_3 and an "ideal" ECI derived by combining the ECIs of C_1, V_1, C_2 in an appropriate way that reflects the action of the curvature process. The distance between two ECIs will be defined in the next section as a function of their correlation coefficient and the length of the corresponding segments. What remains to be decided is how to obtain an ECI corresponding to an "ideal" derived segment that results from C_1, V_1, C_2 due to the production.

If we view the contour segments as fixed-length strings, a curvature process should preserve length and continuity of the curve and its first and second derivative. Therefore, the ideal derived segment should have the same length as the sum of the lengths of C_1, V_1, C_2. Let S_{C_1} and E_{C_2} denote the start and end points of the segments C_1 and C_2, respectively. Continuity of the curve requires that the vector defined by the end points of the ideal segment be equal to the vector from point S_{C_1} to point E_{C_2}. Continuity of the first and second derivative requires that the tangent and curvature of the ideal segment at its endpoints coincide with the tangent and curvature at the points S_{C_1} and E_{C_2}. The sum of the absolute values of ECIs, suggested in a different context in [8], results in a segment that satisfies most of the above desired properties, as stated in the theorem below.

THEOREM 1. *If $\rho_{C1}(\theta)$, $\rho_{V1}(\theta)$, $\rho_{C2}(\theta)$ are the extended circular images of three successive segments of a curve C_1, V_1, C_2, respectively, with C_1, C_2 being convex and V_1 being a concave segment, then their sum,*

$$\rho_{C_1V_1C_2}(\theta) = \rho_{C1}(\theta) - \rho_{V1}(\theta) + \rho_{C2}(\theta), \tag{7}$$

corresponds to a derived convex segment, denoted by $C_1V_1C_2$, that always preserves length and curve continuity, and it preserves continuity of the first and second derivatives of the curve provided that the extended circular image of V_1 is narrower than that of both C_1 and C_2.

Proof. To show preservation of length, we note that the integral of an ECI is equal to the length of the associated curve segment, therefore,

$$\int_{-\infty}^{\infty} \rho_{C_1V_1C_2}(\theta)\, d\theta = \int_{-\infty}^{\infty} \rho_{C1}(\theta)\, d\theta - \int_{-\infty}^{\infty} \rho_{V1}(\theta)\, d\theta + \int_{-\infty}^{\infty} \rho_{C2}(\theta)\, d\theta$$

or

$$\int_{-\infty}^{\infty} \rho_{C_1V_1C_2}(\theta)\, d\theta = L_{C_1} + L_{V_1} + L_{C_2},$$

where L_C denotes the total length of segment C.

Continuity of the curve itself can be shown by focusing on the following integral over a curve segment C,

$$\int_{-\infty}^{\infty} \rho(\theta)\mathbf{t}(\theta)\, d\theta, \tag{8}$$

where $\mathbf{t}(\theta)$ is the tangent unit vector with orientation θ. Since $\rho(\theta) = dl/d\theta$, the integrand can be rewritten as $\mathbf{t}(\theta)\, dl$ or $d\mathbf{l}$, where $d\mathbf{l}$ is the infinitesimal segment of the curve with orientation θ in vector form. Therefore the above integral is equal to the vector from the start to the end point of the curve segment C. Furthermore, the vector from the start to the end point of the ideal segment derived as the result of the production is equal to

$$\int_{-\infty}^{\infty} \rho_{C_1V_1C_2}(\theta)\mathbf{t}(\theta)\, d\theta = \int_{-\infty}^{\infty} \rho_{C_1}(\theta)\mathbf{t}(\theta)\, d\theta - \int_{-\infty}^{\infty} \rho_{V_1}(\theta)\mathbf{t}(\theta)\, d\theta + \int_{-\infty}^{\infty} \rho_{C_2}(\theta)\mathbf{t}(\theta)\, d\theta \tag{9}$$

or

$$\int_{-\infty}^{\infty} \rho_{C_1V_1C_2}(\theta)\mathbf{t}(\theta)\, d\theta = \overrightarrow{S_{C_1}E_{C_1}} + \overrightarrow{S_{V_1}E_{V_1}} + \overrightarrow{S_{C_2}E_{V_2}} = \overrightarrow{S_{C_1}E_{C_2}}, \tag{10}$$

where $\overrightarrow{S_{C_1}E_{C_2}}$ is the vector from the start of C_1 to the end of C_2. Continuity of tangent and curvature is preserved when the domain of $\rho_{V_1}(\theta)$ is narrower than the domain of both $\rho_{C_1}(\theta)$ and $\rho_{C_2}(\theta)$. Under this condition, $\rho_{C_1V_1C_2}(\theta) = \rho_{C_1}(\theta)$ in a right neighborhood of the point $S_{C_1V_1C_2}$ and $\rho_{C_1V_1C_2}(\theta) = \rho_{C_2}(\theta)$ in a left neighborhood of the point $E_{C_1V_1C_2}$. Therefore, the local structure of the original curve is preserved at the endpoints of $C_1V_1C_2$ and thus continuity of first and second derivative is preserved, as well. □

The previous theorem suggests that a plausible way to measure the validity of a hypothesized production $C_1V_1C_2 \rightarrow C_3$, where $C_1V_1C_2$ are consecutive segments of one of the two curves being matched, and C_3 is a segment of the other curve, is to assess the similarity between C_3 and the convex segment derived from C_1, V_1, C_2 by

adding the absolute values of their ECIs. The following definition formalizes this observation.

DEFINITION 4. Given a distance measure between two extended circular images, the *goodness measure* of a production $C_1V_1C_2 \rightarrow C_3$, where C_1, V_1, C_2 are consecutive segments of one of the two curves being matched, and C_3 is a segment of the other curve, is defined as the inverse of the distance between the sum of the absolute values of the ECIs of C_1, V_1, and C_2, and the ECI of C_3.

In the next section we will define a particular distance measure between two extended circular images.

4. THE SHAPE MATCHING ALGORITHM

The intuition behind our matching scheme is that if two closed plane curves correspond to two successive stages of the evolution of shape, then there exist segments that have not deformed to the extent of introducing or eliminating inflection points. The extended circular images of such segments in the two curves have overlapping domains and similar forms. This property can be exploited for establishing correspondences between segments in the two curves. In case inflection points have been introduced or eliminated as a result of the deformation, some segments will remain unassociated. Productions can then be hypothesized to describe the deformation.

Based on the above intuition, our shape matching algorithm relies on an "associate–hypothesize–test" loop. The algorithm first concentrates on the undeformed segments and establishes associations using dynamic programming, while leaving some of the deformed segments unassociated. Already associated segments are then viewed as anchor segments, and they are used to hypothesize productions, which have the potential, if carried out, of making the two curves being matched more similar in shape. Finally, hypothesized productions are tested by computing their goodness measure, and the production with the highest goodness measure is selected and applied.

Application of a production can take place in both directions and results in the replacement in one of the two closed curves of three segments by a single derived segment, whose ECI is the sum of the absolute values of the ECIs of the previous three segments. Application of a producting thus modifies one of the two closed curves being matched, and in fact reduces its number of segments by two. The modified curve hopefully matches the other curve better, since we applied the "best" possible production. At each iteration, the problem becomes progressively simpler, since it involves two fewer segments. The algorithm terminates when there are no unassociated segments, and therefore no productions can be applied.

The matching algorithm does not explicitly compute the derived segments, but represents and uses them in matching via their extended circular images. This is sufficient, since the goodness measure defined previously only depends on the ECIs of the participating segments.

Establishing Associations Using Dynamic Programming

Dynamic programming is an appropriate method for establishing associations between undeformed segments because it can take into account the similarity and overlap of the extended circular images of the segments by using a suitable distance

measure. At the same time it preserves the order of segments along the two contours. The first step in applying a dynamic programming algorithm is to turn the two cyclic sequences of segments into open ones. We perform this step by finding the best matching pair of segments and making these segments the starting segments of the open sequences derived from the cyclic ones. A standard dynamic programming algorithm can then be applied [12] as follows.

Assume that $x = a_1 a_2 \cdots a_n$ and $y = b_1 b_2 \cdots b_m$ are the two curves to be matched, represented as two sequences of segments. We are looking for the minimum cost mapping between x and y that preserves order of segments. The cost of associating a_i with b_j is denoted by $\gamma(a_i, b_j)$ and it is a function of the ECIs of a_i and b_j. We also have to assign a cost to the unassociated segments $\gamma(a_i, \text{nil})$ and $\gamma(\text{nil}, b_j)$, which we take to be a constant.

According to the Wagner and Fischer algorithm [12], we scan x and y from left to right and we build a cost array $D(i, j)$. The (i, j)th element of the array is equal to the optimal cost of matching $x(i) = a_1 a_2 \cdots a_i$, the first i segments of x with $y(j) = b_1 b_2 \cdots b_j$, the first j elements of y. Therefore, $D(i, j)$ can be defined recursively as follows:

$$D(i, j) = \min \begin{bmatrix} D(i-1, j-1) + \gamma(a_i, b_j) \\ D(i-1, j) + \gamma(a_i, \text{nil}) \\ D(i, j-1) + \gamma(\text{nil}, b_j). \end{bmatrix} \tag{11}$$

Finally, we search through the cost array for an optimal path from $D(0,0)$ to $D(n, m)$, which yields the segment associations we are looking for. The complexity of the algorithm is $O(nm)$.

We still have to define $\gamma(a_i, b_j)$, $\gamma(a_i, \text{nil})$, and $\gamma(\text{nil}, b_j)$. We want $\gamma(a_i, b_j)$ to be a function of the following characteristics: overlap and similarity of the ECIs of segments a_i and b_j, and closeness of the lengths of a_i and b_j. Overlap and similarity of the ECIs is captured by the distance metric d_1, where

$$\frac{1}{d_1} = \frac{\int_{-\infty}^{\infty} \kappa_{a_i}(\theta)\kappa_{b_j}(\theta)\, d\theta}{\sqrt{\int_{-\infty}^{\infty} \kappa_{a_i}^2(\theta)\, d\theta \int_{-\infty}^{\infty} \kappa_{b_j}^2(\theta)\, d\theta}} \tag{12}$$

where $\kappa(\theta) = 1/\rho(\theta)$ is the inverse ECI. It can be easily seen that d_1 is equal to 1 if $\kappa_{a_i}(\theta) = \lambda \kappa_{b_j}(\theta)$. If $\kappa_{a_i}(\theta)$ does not overlap with $\kappa_{b_j}(\theta)$, then d_1 is equal to ∞, because the numerator is equal to zero for all θ. This distance metric is insensitive to absolute values of curvature, therefore it is size independent. Finally, it takes into account the exact shapes of the two ECIs, and produces the shortest distance in the case of both maximum overlap of the two ECIs and greatest similarity in shape. The integrals are approximated by summations in the case of a discrete implementation of ECIs.

The above distance metric alone fails to capture directly the difference between the lengths of the two segments. Therefore it may result in a small distance if the domain of one ECI is much shorter and completely contained in the other. To take

into account length differences, we introduce an additional distance factor d_2,

$$\frac{1}{d_2} = 1 - \frac{||a_i| - |b_j||}{|a_i| + |b_j|}, \tag{13}$$

where $|a_i|$, $|b_j|$ denote the lengths of the corresponding segments. d_2 is 1 if the two arcs are identical in length and very large if one is much longer than the other.

Finally, we define:

$$\gamma(a_i, b_j) = d_1 d_2. \tag{14}$$

Thus the final distance measure weighs equally the overlap and similarity of the two ECIs and similarity of the two segments in terms of length. $\gamma(a_i, b_j)$ is between 1 and ∞. The cost of a non-match should be set to a value low enough to make unassociation preferable to association of the wrong segments but not too low so that right associations are missed.

A final issue to resolve regarding ECIs is the periodic nature of the unit circle and its interaction with rotation indices absolutely greater than 1. If a curve spirals by more than a full circle, then the mapping from tangent direction to the unit circle is not one-to-one any more. In the implementation of our matching algorithm, we deal with this problem by treating ECIs as non-periodic functions, extending from $-\infty$ to ∞, with an ambiguity of $2\kappa\pi$ in their argument. In computing the distance d_1, we select κ, so as to maximize the overlap between $\kappa_{b_j}(\theta + 2\kappa\pi)$ and $\kappa_{a_i}(\theta)$, and we correlate the latter two inverse extended circular images.

Process Recovery

The previous section showed how to establish associations between similar segments of the two shapes. In this section, we concentrate on the unassociated segments and we attempt to recover the productions related to them.

The two classes of productions in the process grammar naturally introduce two levels of abstraction in process recovery. The first deals with segments of constant curvature sign and recovers segment-based continuation processes of the $CVC \rightarrow C$ or $VCV \rightarrow V$ type, as discussed earlier. The second level deals with curvature extrema within pairs of segments and recovers bifurcation processes by concentrating on the extrema of the extended circular images of associated segments. In this paper we only address recovery of segment-based continuation processes, bifurcation processes being dependent on point primitives and therefore highly unstable, as shown in Fig. 3.

The algorithm for recovering segment-based continuation processes takes as input an ordered list of pairs of the form (a_i, b_j) or (a_i, nil) or (nil, b_j), i.e., the output of the dynamic programming algorithm. It then generates all possible productions in both directions of the form $C_1VC_2 \rightarrow C_3$ or $V_1CV_2 \rightarrow V_3$ for which the following conditions are satisfied:

- one of the three segments of the left-hand side of the production is associated with the right-hand side.
- the two other segments of the left-hand side of the production are unassociated.

The first condition ensures that we hypothesize productions that may improve existing associations. The second condition ensures that productions only absorb unassociated segments.

All productions thus obtained are ordered according to a goodness measure, equal, for a production $C_1VC_2 \rightarrow C_3$ with prior association of C_1 with C_3, to the ratio,

$$\frac{\text{distance}(C_1VC_2, C_3)}{\text{distance}(C_1, C_3)} \tag{15}$$

where C_1VC_2 denotes the segment whose ECI is equal to the sum of the absolute values of the ECIs of C_1, V, and C_2. The "best" production is selected (regardless of direction) and it is effected, i.e., the three segments C_1, V, and C_2 are replaced by their combination C_1VC_2, described by its extended circular image and the sign of its curvature, in this case positive. Thus the closed curve containing C_1, V, and C_2 gets modified, and the modified curve has two segments less than the original. The modified curve is again described by its extended circular image. As mentioned above, new correspondences are established between the modified curve and the closed curve containing C_3, and the new cost is compared with the cost of the

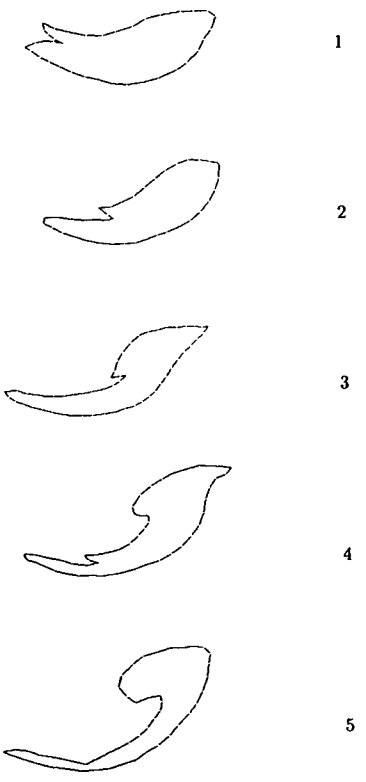

FIG. 4. Cloud silhouettes hand-traced from real infrared satellite images. A leaf-shaped cloud evolves into a comma-shaped cloud during cyclogenesis.

original associations. If the new cost is lower than the original cost, the iteration is repeated by selecting a new "best" production. If the new cost is higher than the original cost, then the iteration terminates by returning a match that includes one or more unassociated segments.

5. EXPERIMENTAL RESULTS

In this section, we present experimental results from the application of shape matching to the analysis of sequences of cloud silhouettes, such as those appearing in visible or infrared satellite imagery. Such sequences contain a lot of information about evolving weather patterns, and therefore they are very useful in weather forecasting. In this domain, it is important to describe evolving cloud silhouettes, and eventually match such descriptions against knowledge bases of various meteorological phenomena. Another major issue that we do not address her is the extraction of cloud silhouettes from real satellite imagery. Recent progress in curve detection [30] and perceptual grouping [26, 29] is very promising in regard to curve extraction problems from real images.

Figure 4 shows cloud silhouettes from the evolution of a storm over North America, that were hand-traced from real satellite images. The storm pattern starts

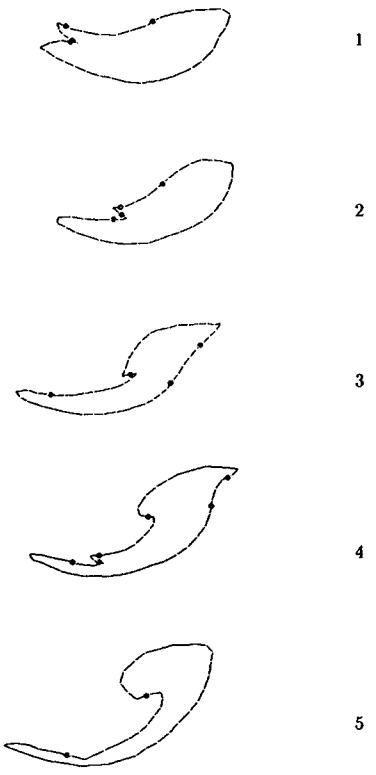

FIG. 5. Segments computed for the cloud data of the previous figure. Segments are bounded by inflection points shown as dots. For easy reference, convex or concave segments are denoted by $C(a)$ or $V(a)$, respectively, where a is the length of the segment, measured in units equal to the horizontal distance between neighboring pixels.

as a leaf shape and gradually turns into a comma shape, with corresponds to a mature storm. The sampling period of our example is 3 h whereas in reality a new satellite image is available every 30 min.

Figure 5 also shows the significant inflection points detected by the preprocessing step of our algorithm. Each of the segments is named by its length for reference purposes. Figure 6 shows the productions recovered by the matching step of the

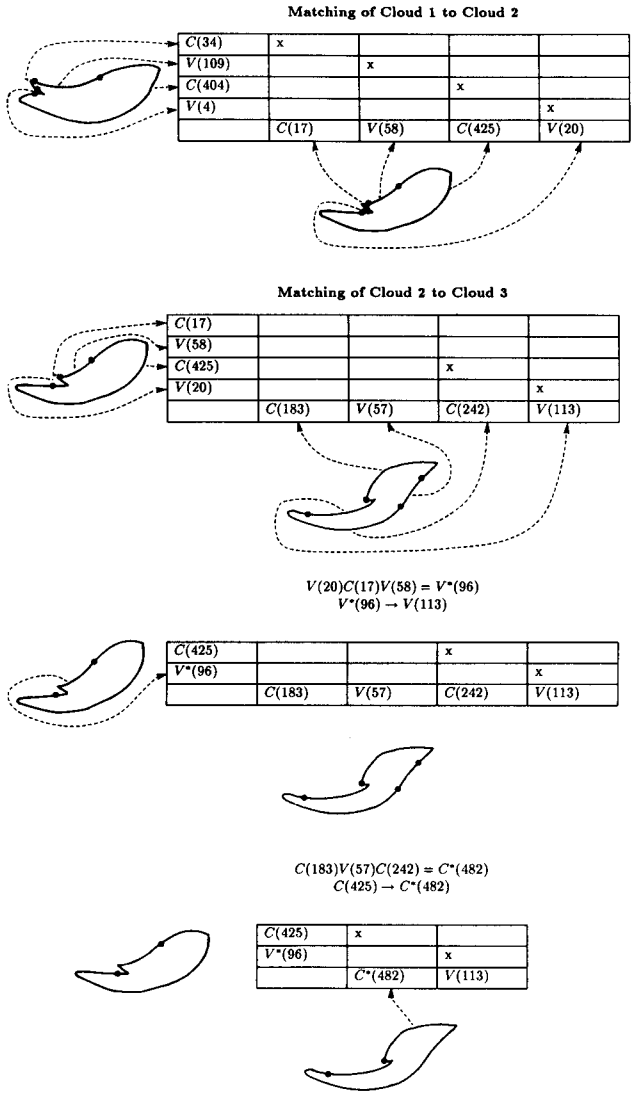

FIG. 6. The trace of the matching algorithm is shown on the successive pairs of the cloud sequence. Dotted arrows associate table entries with the corresponding cloud segments. Successive iterations are ordered top–down. In iterations other than the first (the top one), only the derived segment is associated with its table entry with a dotted arrow. Note that the "associate" step of the algorithm correctly associated the derived segment with the single segment occupying the other side of the selected production of the previous cycle. The figure continues on the next page.

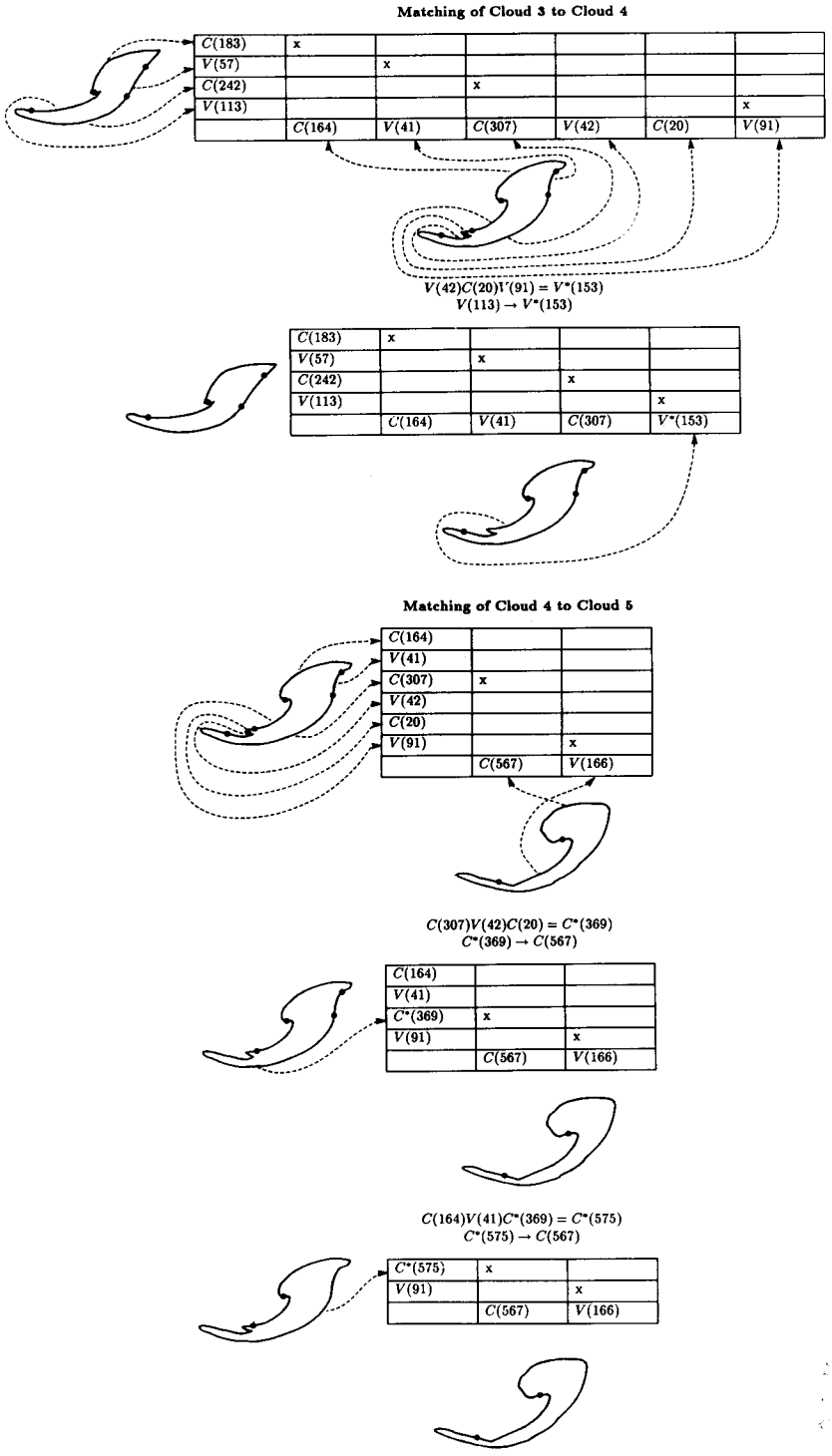

FIG. 6—*Continued*

algorithm for successive pairs of clouds. Matching Cloud 1 with Cloud 2 did not require any productions, since segments matched each other exactly. Cloud 2 and Cloud 3 both have the same number of segments. However, two productions are required (in opposite directions) to account for their differences. Matching of Cloud 3 and Cloud 4 requires one production to account for an extra protrusion present in Cloud 4. Finally, matching of Cloud 4 with Cloud 5 requires two productions, both in the same direction, to account for the fact that Cloud 4 has four more segments than Cloud 5. It is worth noting that the segment $C^*(369)$ derived from the first production takes part in the second production, as well. This underscores the notion that derived segments are treated like original segments by the matching algorithm. Finally note that in all examples, the dynamic programming algorithm, when run on the segments resulting from a production application, always matches the newly derived segment with the segment on the other side of the production.

The productions recovered by the matching algorithm can then form the basis for an interpretation of this cloud sequence. In meteorology, the absolute orientation of individual segments matters, and this information is captured by extended circular images. For example, the leaf-shaped pattern usually has an east–west orientation, with the tail of the leaf facing west. This is a result of the wind direction in the northern hemisphere.

If N is the number of points used to represent the input shapes, and M is the number of segments found in the two input shapes, then the computational complexity of the matching algorithm is:

- Preprocessing step (spline modelling and extraction of significant inflection points) is $O(N)$.
- Each iteration of the algorithm includes a dynamic programming step, which is $O(M^2)$, and a production recovery step, which is $O(M)$. The maximum total number of iterations is $O(M)$, therefore the complexity of the iterations is $O(M^3)$.
- The cost of the correlation of extended circular images is linear with the number of sampling points we use. This is constant with respect to the complexity of the input shapes.

6. CONCLUSIONS

In this paper we presented a new segmented representation for closed planar curves based on the concept of the extended circular image (the 2-dimensional analog of the extended Gaussian image). A closed curve is represented as a cyclic sequence of the extended circular images of its segments delimited by inflection points. The extended circular image is parameterized by the angle θ formed by the tangent (or the normal) of the contour and the x axis, in contrast to the commonly used arc length parameterization. The most important property of a parameterization based on tangent orientation is that it encodes information about absolute orientation of the segment, which can then be exploited directly in matching. Length is implicitly encoded in the extended circular image as its integral over θ. Another advantage of using tangent orientation instead of arc length is that two curves with the same shape but different size result in representations that are identical within a scale factor. Representation based on arc length have different domains, requiring adjustment of the scale of their independent variable. As a result, extended circular

images of two segments pointing to approximately the same direction can be directly correlated, whereas the same is not true in an arc length parameterization.

Absolute orientation of convex and concave segments is one major characteristic we use in matching. The other is the order in which the segments appear along the contour. To produce non-crossing associations that preserve the order of segments we use dynamic programming. The overall matching algorithm iterates between establishing segment associations using a dynamic programming algorithm and effecting the most plausible production that involves unassociated segments next to two associated ones. This greedy algorithm converges when all segments have been associated or when no production can be found that reduces the overall cost.

It is worth noting that our algorithm depends on a single threshold, the cost assigned to unassociated segments in the dynamic programming matching algorithm. Experiments show that the algorithm is robust with respect to this threshold, in that it gives the same answer for a wide range of values, as long as the threshold remains higher that the cost of association of perceptually similar segments and lower than the cost of association of spuriously overlapping extended circular images.

APPENDIX: COMPUTING CURVATURE OF POLYGONAL PLANE CURVES VIA SMOOTH APPROXIMATIONS

Digitized curves are usually available in the form of discrete sample points. Connecting such points by straight line segments leads to polygonal plane curves. Polygonal plane curves have been used extensively in shape analysis as the basis for syntactic pattern recognition approaches [14], and for curve encoding via split-and-merge approximation to obtain other polygonal approximations with fewer line segments that capture the general characteristics of the curve.

Many earlier approaches to curvature computation used piecewise linear approximations [19, 6]. However, curvature is a function of the first and second derivatives of a curve with respect to a curve parameter, and therefore using the angle between successive line segments joining the sample points of the curve may result in unreliable measures of curvature. Higher order approximations to the curve appear to be preferable in that respect, and they can combine smoothing or regularizing effects [18] with accurate derivative computations obtained analytically.

In this paper, the objective is not to study optimal curvature computation, therefore we use a standard local B-spline approximation, that is widely used in computer graphics [20]. B-splines are approximating splines, which means that they do not pass through the input curve points (called knots), contrary to local interpolating splines. This is the price to pay in B-splines for obtaining continuity of the first and second derivatives of the spline approximation. Interpolating splines are discontinuous in the second derivative and therefore discontinuous in curvature. A property of B-splines is that they lie within the convex hull of their knots. This is not true of interpolating splines, which can demonstrate kinks and wild fluctuations. Another whole class of splines are global splines, which fit a single cubic polynomial to all input points. The utility of such splines in practice is limited, because they may display arbitrary kinks and twists. Local spline approximation fits a pair of cubic polynomials to every successive four points of the curve. The resulting piecewise polynomial curve is continuous in its first and second derivation everywhere.

The classic local cubic B-spline is computed for four points at a time and is a cubic curve in a single parameter u. Assume four points A_{-1}, A_0, A_1, A_2, with (x, y) coordinates equal to (x_i, y_i), for $i = -1, 0, 1, 2$. The local cubic B-spline for these four points is of the form

$$X(u) = A_x u^3 + B_x u^2 + C_x u + D_x = [u^3, u^2, u, 1][A_x, B_x, C_x, D_x]^T \quad (16)$$

$$y(u) = A_y u^3 + B_y u^2 + C_y u + D_y = [u^3, u^2, u, 1][A_y, B_y, C_y, D_y]^T \quad (17)$$

where $(X(u), Y(u))$ are the coordinates of the spline points, for $0 \leq u \leq 1$. The coefficient vectors are computed from the coordinates of the four points in the following manner:

$$[A_x, B_x, C_x, D_x]^T = B[x_{-1}, x_0, x_1, x_2]^T \quad (18)$$

$$[A_y, B_y, C_y, D_y]^T = B[y_{-1}, y_0, y_1, y_2]^T, \quad (19)$$

where B is a 4×4 matrix:

$$B = \frac{1}{6}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}. \quad (20)$$

The matrix B has been computed so as to satisfy continuity of the spline and its first and second derivative at the boundary points, namely for $u = 0$ and $u = 1$ [20]. Figure 7 shows an example of a B-spline approximation to a small number of knot points.

In contrast to computer graphics, where the resulting spline approximation is sampled to generate an interpolated version of the original curve, we are interested in using splines to perform computations of quantities such as curve length,

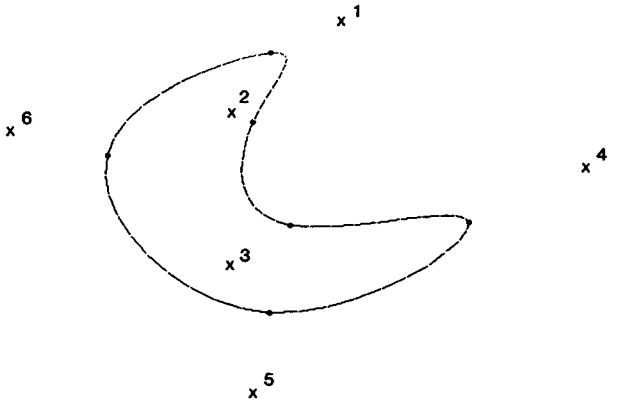

FIG. 7. An example of a piecewise polynomial curve consisting of local B-splines. Spline endpoints are shown as dots. Knot points are shown by "x" symbols. Knot points are numbered to indicate their order.

curvature extrema, and zero crossings and distance between a point and a spline. Such computations in general do not have analytical solutions, therefore the most convenient representation of the original piecewise linear curve is as a sequence of cubic splines in their analytic form, or equivalently as a sequence of pairs of 4-vectors corresponding to the spline coefficients. We can thus carry out numerical solutions to these problems on demand.

Curvature Computation

The curvature κ for any curve which is continuous in its first and second derivatives and parameterized by u is given by the formula

$$\kappa(u) = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{(\dot{x}^2 + \dot{y}^2)^{3/2}} \equiv \frac{w}{z^{3/2}}, \tag{21}$$

where derivatives are with respect to u. Note that w is a second degree polynomial in u (the third-order terms cancel out) and z is a fourth-degree polynomial.

Curvature extrema of a piecewise B-spline approximation to a curve can be of two types: internal to a local spline or on the boundary between two local splines. In the interior of a spline ($0 < u < 1$), the curve is infinitely continuous and differentiable, therefore internal curvature extrema are characterized by $d\kappa(u)/du = 0$, or

$$\dot{w}z - \tfrac{3}{2}w\dot{z} = 0. \tag{22}$$

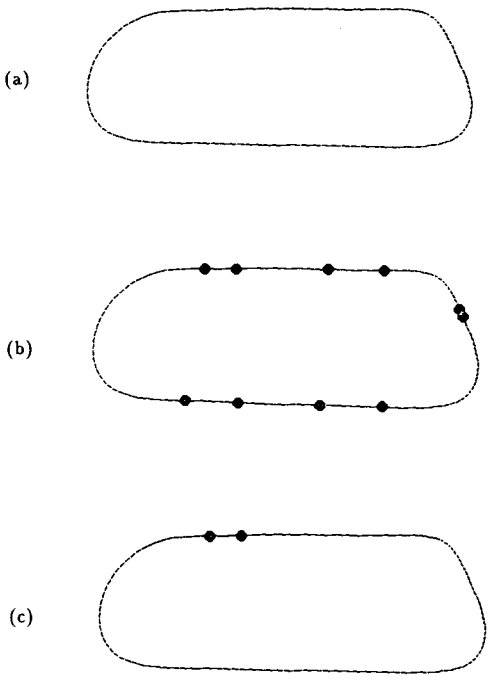

FIG. 8. Flat segments are characterized by narrow extended circular images: (a) a closed curve; (b) its inflection points; (c) result of merging flat segments using productions. Note that the remaining two segments cannot be reduced to one using productions.

Solution of this fifth-degree polynomial equation can be performed numerically (using, for example, the bisection or Newton–Raphson method [16]). On the boundary between two splines, curvature is continuous but not its derivative with respect to u. To check whether a boundary corresponds to a curvature extremum, it is sufficient to check the sign of $d\kappa(u)/du$ at the two adjoining ends. Curvature zero crossings or inflection points are computed by solving the equation $\kappa(u) = 0$, which is reduced to the quadratic equation $w(u) = 0$.

Treatment of Flat Segments

A flat segment is characterized by a narrow extended circular image and low average curvature. A flat segment may arise if curvature remains low in absolute value and crosses zero before a significant change in the orientation of the tangent has occurred (Fig. 8). To prevent flat segments from complicating the matching task, we replace triples of consecutive segments with a single segment by applying a $CVC \rightarrow C$ or a $VCV \rightarrow V$ production, if the middle segment has a narrow extended circular image. A closed convex curve cannot be obtained by this approach, since we always start with an even number of segments, and each production reduces three segments to one. Eventually, two segments will remain, which cannot be reduced to one, unless we allow for $C_1 = C_3$ in a production $C_1V_1C_2 \rightarrow C_3$.

ACKNOWLEDGMENTS

The author is grateful to Mike Leyton (whose work on process grammars was the initial inspiration for this work), Mike Luby, and Geoff Hinton for helpful discussions. John Tsotsos' comments on an early draft of this paper and Mike Leyton's comments on a late one helped significantly improve both presentation and technical content of the paper. John Tsotsos' suggestion that dynamic analysis of clouds would be an interesting application and a brainstorming session with Louis Garand and Pat King of Environment Canada helped provide a real context for this work. Johnny Amanatides provided help with the local B-spline literature. Training manuals, from which the cloud silhouette data was extracted, were kindly provided by the Atmospheric Services Division of Environment Canada in Toronto. The author was financially supported by a Strategic Grant from the Natural Sciences and Engineering Research Council of Canada and a Grant from the Government of Ontario through the Information Technology Research Centre.

REFERENCES

1. H. Asada and M. Brady, The curvature primal sketch, *IEEE Trans. Pattern Anal. Mach. Intell.* **8**, No. 1, 1986, 2–14.
2. F. Attneave, Some informational aspects of visual perception, *Psychol. Rev.* **61**, 1954, 183–193.
3. P. Bouthemy and A. Benveniste, Modeling of atmospheric disturbances in meteorological pictures, *IEEE Trans. Pattern Anal. Mach. Intell.* **6**, No. 5, 1984, 587–600.
4. M. Brady and H. Asada, Smoothed local symmetries and their implementation, *Int. J. Robot. Res.* **3**, No. 3, 1984, 36–61.
5. L. Davis, Shape matching using relaxation techniques, *IEEE Trans. Pattern Anal. Mach. Intell.* **1**, (1979), 60–72.
6. L. Davis, Understanding shape: Angles and sides, *IEEE Trans. Comput.* **26**, No. 3, 1977, 236–242.
7. D. Hoffman and W. Richards, Parts of recognition, *Cognition* **18**, 1985, 65–96.
8. B. Horn, *Robot Vision*, MIT Press, Cambridge, MA, 1986.
9. M. Kass, A. Witkin, and D. Terzopoulos, Snakes: Active contour models, in *1st Int. Conf. Computer Vision*, 1987, pp. 259–268.
10. M. Leyton, A process grammar for shape, *Artif. Intell.* **34**, No. 2, 1988, 213–247.

11. J. Little, An iterative method for reconstructing convex polyhedra from external images, in *Proceedings, American Association for Artificial Intelligence Conference, 1983*, pp. 247–254.
12. L. Miclet, *Structural Methods in Pattern Recognition*, Springer-Verlag, New York, 1986.
13. R. Millman and G. Parker, *Elements of Differential Geometry*, Prentice–Hall, Englewood Cliffs, NJ, 1977.
14. T. Pavlidis, Algorithms for shape analysis of contours and waveforms, *IEEE Trans. Pattern Anal. Mach. Intell.* **2**, No. 4, 1980, 301–312.
15. T. Pavlidis, A critical survey of image analysis methods, in *8th IEEE Int. Conf. on Pattern Recognition, 1986*, pp. 502.
16. G. Phillips and P. Taylor, *Theory and Applications of Numerical Analysis*, Academic Press, London, 1973.
17. W. Richards, J. Koenderink, and D. Hoffman, *Inferring 3D shapes from 2D codons*, Technical Report AIM-840, MIT AI Lab, 1985.
18. B. Shahraray and D. Anderson, Optimal smoothing of digitized contours, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1986*, pp. 210–218.
19. L. Shapiro, A structural model of shape, *IEEE Trans. Pattern Anal. Mach. Intell.* **2**, 1980, 111–126.
20. A. R. Smith, *Spline Tutorial Notes*, Technical Report Memo 77, Lucasfilm Ltd., 1983 (rev. 1984).
21. J. Tsotsos, J. Mylopoulos, D. Covvey, and S. Zucker, A framework for visual motion understanding, *IEEE Trans. Pattern Anal. Mach. Intell.* **2**, No. 6, 1980, 563–573.
22. D. Walters, Selection of image primitives for general-purpose visual processing, *Comp. Vision Graphics Image Process.* **37**, No. 3, 1987, 261–298.
23. R. Weldon, *Cloud Patterns and the Upper Air Wind Field* (training course notes), Technical Report 4, National Environmental Satellite Service, U.S. Department of Commerce, 1979.
24. A. Witkin, D. Terzopoulos, and M. Kass, Signal matching through scale space, in *Proceedings American Association for Artificial Intelligence Conference, 1986*, pp. 714–719.
25. M. Leyton, symmetry-curvature duality, *Comput. Vision Graphics Image Process.* **38**, 1987, 327–341.
26. D. Lowe, Organization of smooth image curves at multiple scales, in *Proceedings, International Conference on Computer Vision*, 1988, pp. 558–567.
27. W. Martin and J. Aggarwal, Computer analysis of dynamic scenes containing curvilinear figures, *Pattern Recognit.* **11**, 1979, 169–178.
28. F. Mokhtarian and A. Mackworth. Scale-based description and recognition of planar curves and two-dimensional shapes, *IEEE Trans. Pattern Anal. Mach. Intell.* **8**, No. 1, 1986, 34–43.
29. A. Sha'ashua and S. Ullman, Structural saliency: The detection of globally salient structures, *Proceedings, International Conference on Computer Vision*, 1988, pp. 321–327.
30. S. Zucker, C. David, A. Dobbins, and L. Iverson, The organization of curve detection: Coarse tangent fields and fine spline coverings, *Proceedings, International Conference on Computer Vision*, 1988, pp. 568–577.