Computing 2D Motion of Boundaries from Correspondences of Points of Significant Curvature

T. Rachidi
School of Science and Engineering
Al-Akhawayn University in Ifrane
PO. BOX 104, Ifrane, 53 000
Morocco

L. Spacek
Department of Computer Science
University of Essex
Colchester, CO4 3SQ
U.K

Abstract

Finding the parameters of 2-D motion of curves is posed as a correspondence problem between points of significant curvature along these curves. The algorithm developed for this purpose is simple and intuitive. It uses natural ordering of such points along curves, the consistency of the expansion factor, and the angle of rotation to constrain the search for the true pairings. The parameters of the motion – namely the expansion, translation and rotation - are then evaluated from the positions of matched points of significant curvature rather than from curvature values as done by other techniques. This novel technique is robust and copes particularly well with occlusion problems at curve ends. In particular, our technique is shown to be more accurate in determining the parameters than the Hough Transform.

Keywords: 2D Motion, Correspondence, high curvature points, Affine Transform, Boundaries, Curves.

1 Introduction

An important task in computer vision and image processing is to determine the transformation which maps a *2-D* curve into another. Such transformations are of crucial importance to *2-D* recognition [4, 13, 2], motion and tracking. To this end, points of significant curvature have received special attention in the computation of the parameters of the transformation due to their local nature. A comprehensive comparison of curvature estimation methods has been reported in [15]. More recently, the results of a series of experiments on three different estimators of point curvature in varying degrees of noise were published [6].

Many methods have been proposed to evaluate the parameters of the transformation of a *2-D* curve into another. Such techniques can be classified into local and global: the latter are based on matching a global vector of features, and thus suffer from occlusion [12]; the former use local features (such as curvature), which depend only upon portions of objects [1, 7].

Cohen *et al.*, proposed a method for determining transformations between two curves. This method is based on the minimisation of energy which tends to preserve the matching of points of significant curvature, while ensuring a smooth field of displacement vectors everywhere [3]. However, this technique is suited for deformable curves, and is very expensive since it requires the solution of a second order partial differential equation. Furthermore, the method also requires a first guess to start the iterative solution of the differential equation, and does not explicitly solve for the parameters of the transformation.

Ma and Chen's technique is based on a Generalised Hough transform [8], and can be summarised as follows:

For a given pair of boundaries M and its transform N, let $(m, n) \in M \times N$ be a pair of corresponding points on the continuous curves M and N. Asserting that $m = (x(t), y(t))^T$ corresponds to $n = (u(t), v(t))^T$ gives:

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = kR \begin{bmatrix} u(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \quad (1)$$

where k is the expansion factor, $X_0 = (x_0, y_0)^T$ is the translation vector and R is the rotation matrix:

$$R = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$$

The slope and curvature at a point $m = (x(t), y(t))^T$ on a continuous curve parametrised by t are given by:

$$tg(\alpha) = \frac{y'}{x'} \quad (2)$$

$$\zeta = \frac{x'y'' - y'x''}{(x'^2 + y'^2)^{\frac{3}{2}}} \quad (3)$$

where $x' = dx/dt$, $x'' = d^2x/dt^2$, $y' = dy/dt$ and $y'' = d^2y/dt^2$.

After some elementary mathematics, the slope angles α_i and β_i on corresponding points m_i and n_i are related by:

$$tg(\alpha_i - \beta_i) = tg(\theta) \quad (4)$$

Additionally, the curvatures ζ_i and γ_i on corresponding points m_i and n_i obey the equation:

$$\zeta_i = \frac{1}{k}\gamma_i \quad (5)$$

Ma and Chen's algorithm is as follows:

1. Form two two-dimensional accumulator arrays $A(X_0)$ and $B(k, \theta)$,
2. For every pair of points $(m_i, n_i) \in M \times N$, compute (k, θ, x_0, y_0) using the above equations, and increment $A(X_0)$ and $B(k, \theta)$.

Local maxima in $A(X_0)$ and $B(k, \theta)$ correspond to the parameters of the transformation.

There are a number of intrinsic problems with this approach:

- the complexity of the method is given by the size $M \times N$.
- due to the sensitivity of curvature to perturbations and noise, the obtained values for k (ratio of curvatures) lack precision;
- the approach does not cope well with curves containing points where curvature is undefined, such as perfect corners (see **Figure** 1). This is due to the fact that curvatures are the same at all corners $P1$, $P'1$, $P2$, and $P'2$.

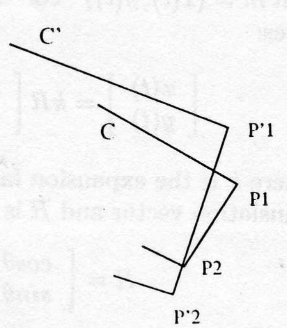

Figure 1. *The curve C' was obtained by rotating and expanding the curve C.*

This problem, actually, reflects the locality of the technique. In other words, through a small aperture placed over the corner, as shown in **Figure** 2, it is impossible to say if an expansion is occurring or not. Of course, if there is another point of high curvature within the aperture, then the combination of local measurements at these points can yield the expansion rate. This is, the aperture problem [14].

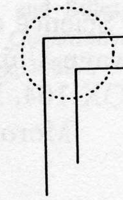

Figure 2. *The aperture problem, looking at the expanding boundary through a small aperture, it is impossible to determine the expansion factor at corners.*

2 Establishing Motion Parameters

In perceiving the motion of **Figure** 1, it appears that the human visual system uses the distances between corners of each curve to work out the expansion, and uses the orientation of the lines joining these corners to derive the rotation[1]. Looking at the locations of all points of significant curvature within the same boundary is a global approach which overcomes the aperture problem.

Based on this remark, the parameters of the transformation can be defined as follows:

$$k = \frac{\|P'1\vec{P}'2\|}{\|P1\vec{P}2\|} \quad (6)$$

$$\theta = acos\left[\frac{P'1\vec{P}'2.P1\vec{P}2}{\|P'1\vec{P}'2\|.\|P1\vec{P}2\|}\right] \quad (7)$$

$$X_0 = P'1 - kR.P1 \quad (8)$$

where k is the expansion factor, θ the angle of rotation, and X_0 the translation vector.

What seems important then is not so much the value of the curvature at the corners, but the locations of these corners – points of maximum curvature, in general. It follows that, if correspondence between high curvature points of C and C' is estabilsed, the paremeters k, θ, and X_0 can be computed using the above equations.

[1] We have no proof for these claims. This matter may be a fruitful topic of research in psychology, in which measurements of perception of expansion and rotation using curvature on the one hand, and positions of high curvature points, on the other hand, are compared.

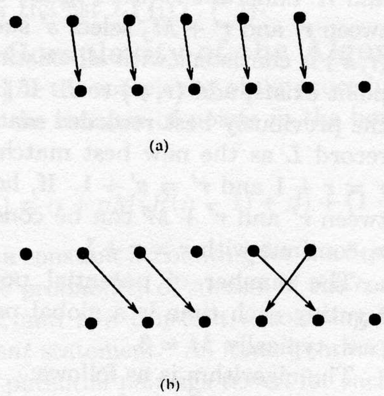

Figure 3. *Bullets designate significant curvature points. Arrows indicate pairings. (a) allowed configuration. (b) disallowed configuration because of the crossing.*

3 Moving the Goal Post

With the above in mind, finding the parameters of motion is posed as a matching problem in this work, that is, establishing correspondence between points of significant curvature. The actual evaluation of the parameters is obtained from the resulting pairings.

This matching must be a one-to-one, and without crossings (see **Figure** 3). The first constraint (*i.e.,* one-to-one) stems from the physical nature of points of high curvature, *i.e.,* a point of high curvature in the scene always projects into a unique point in the image plane. The second constraint (*i.e.,* prohibition of crossings) is dictated by the natural ordering on boundary points. The ordering not only allows straightforward processing and measurements, but must also be satisfied by all plausible solutions.

Additionally, if such matching exists for a solid boundary undergoing a transformation, consistency must be conserved between pairs of corresponding high curvature points. Let $P_1, P_2, ...P_n$ be the ordered points on C, and let $P'_1, P'_2, ...P'_n$ be their corresponding high curvature points on C'. Let

$$k_{i,j} = \frac{\|\vec{P'_i P'_j}\|}{\|\vec{P_i P_j}\|} \qquad \forall i,j \in \{1..n\}, \quad i \neq j \tag{9}$$

$k_{i,j}$ represents the expansion that the curve C undergoes to become C', estimated here from P_i, P'_i, P_j and P'_j. Because of the rigidity assumption, the expansion factor is the same for all pairs of corresponding points (see **Figure** 4).

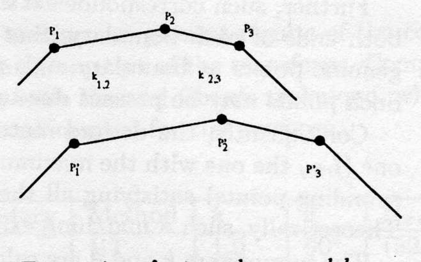

Figure 4. *Expansion factors* $k_{1,2}$ *and* $k_{2,3}$ *are consistant* i.e., $k_{1,2} = k_{2,3}$. *Similarly the angles of rotation are equal* i.e., $\theta_{1,2} = \theta_{2,3}$.

Thus, the consistency constraint may be written as follows:

$$\forall i,j \in \{1..n\}, \ i \neq j, \ \forall h,l \in \{1..n\}, \ h \neq l,$$

$$k_{i,j} = k_{h,l} \tag{10}$$

In practice, such a constraint is very strong, and has to be relaxed to allow for small differences between the different $k_{i,j}$. Moreover, it is not necessary to consider all pairs of points $P_i, P'_i, P_j, P'_j, P_h, P'_h, P_l$ and P'_l. It can be shown that the subset of successive points $P_i, P'_i, P_{i+1}, P'_{i+1}, P_{i+2}$ and P'_{i+2} gives a sufficient constraint. In the light of the above remark, the consistency constraint is then expressed as follows:

$$\forall i \in \{1..n-2\}, \quad |k_{i,i+1} - k_{i+1,i+2}| \leq \epsilon_1 \tag{11}$$

where ϵ_1 represents the maximum difference allowed. Typically $\epsilon_1 = 0.05$.

This constraint is computationally less expensive than (10), since only a subset of all the possible combinations of pairs of points, satisfying the ordering constraint, is considered.

Similarly, a second constraint can be derived for the angle of rotation θ. Let $\theta_{i,j}$ represent the angle of rotation that curve C undergoes to become C', estimated from P_i, P'_i, P_j and P'_j. Because of the rigidity assumption, this angle is the same for all pairs of corresponding points. Thus, the second consistency constraint can be written as follows:

$$\forall i \in \{1..n-2\}, \quad |\theta_{i,i+1} - \theta_{i+1,i+2}| \leq \epsilon_2 \tag{12}$$

where ϵ_2 represents the maximum difference allowed in the angle of rotation. Typically $\epsilon_2 = 5°$.

The two constraints above (11) and (12) not only reduce the complexity of the process by a great deal, but also categorically discard wrong points (*i.e.* those which constitute noise and digitisation effect), since such points would violate the constraints.

Further, such correspondence should be "loose" at both ends of each boundary, that is, it should allow genuine points at boundary ends not to be matched. Such points may be present due to occlusion.

Consequently, the desired matching is the longest one (*i.e.*, the one with the maximum number of corresponding points) satisfying all the above constraints. Theoretically, such a matching exists and is unique.

The parameters k and θ are calculated respectively from $k_{i,i+1}$, and $\theta_{i,i+1}$ of the final matching as follows:

$$k = \frac{1}{n-1}\sum_{1}^{n-1} k_{i,i+1} \quad \text{and} \quad \theta = \frac{1}{n-1}\sum_{1}^{n-1} \theta_{i,i+1} \qquad (13)$$

4 Constrained Point Matching

Curvature profiles are computed from cubic Spline approximations of each curve. However, it is not necessary to compute the curvature. Only the locations of points of maximum curvature are of interest here. Points of significant curvature for each curve are then stored in arrays R and R' for processing.

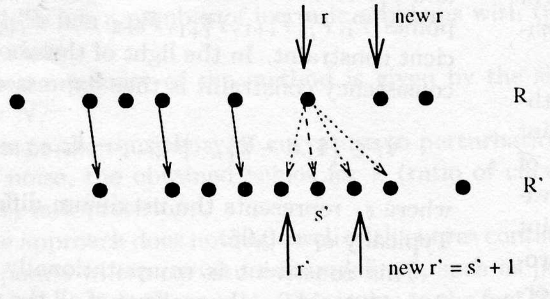

Figure 5. *Establishing correspondence between points. Point r is currently under examination. Among the M possible pairings (marked with dotted arrows), s' (marked with a dashed arrow) is the pairing consistent with the previously established pairs (marked with solid arrows). The search continues with the point marked new r under focus, and with r' = s' + 1.*

Note that at least two significant points must be present in C and in C'. Curves which do not satisfy this condition are called "simple" curves, and are ignored.

To find the longest match satisfying the above constraints, an exhaustive search is employed. Progressing from left to right, points from R and R' are tested in the following manner: let $L = \{(p, p')\}$ be the list of points matched at a given stage; by construction, pairs in L satisfy the above conditions. Initially $L = \emptyset$. Let r and r' be the indices of the current points of R and R' under focus (see **Figure** 5). For all points between r' and $r' + M$, select s' such that the new pair (r, s') is consistent with all existing pairs in L. If such point exists, add (r, s') to L. If $\sharp L > \sharp B$, (where B is the previously best recorded match, and $\sharp B$ its size), record L as the new best match, and continue with $r = r + 1$ and $r' = s' + 1$. If, however, no point between r' and $r' + M$ can be consistently paired with r, continue with $r = r + 1$.

The number of potential points from R' under scrutiny each time is a global parameter of the process, typically $M = 3$.

The algorithm is as follows:

```
Search(R,N,r,R',N',r',L,l)

ARRAYS of PEAKS R,R'
LENGTHS N,N'
INDICES r,r'
LIST of PAIRS L
LENGTH l

BEGIN
    IF (r ≥ N  OR  r' ≥  N') THEN
        RETURN;

    FOR i=r' TO r'+M DO
      IF Consistent(R[r],R'[i],L,l) THEN
        L ← L ∪ {(r,i)} ;
        Search(R,N,r+1,R',N',i+1, L, l+1);
        IF Improvement(L,B) THEN
           B = L;
        END
      END
    END
END
```

Algorithm 4.1. *Algorithm for Constrained Point Matching.*

Improvement(L,B) determines whether there is an improvement in the pairings of L with respect to the best recorded pairings at the time of the call. Clearly, if L contains more pairings than B, an improvement has occurred. If, however, the lists B and L have the same number of pairs, L is said to be better than B if the pairings in L give a smaller rotation angle than the pairings in B. This only happens when there is an intrinsic ambiguity in the transform.

The function *Consistent(P,P',L,l)* returns 1 if the pair (P, P') is consistent with the list of pairings L (of length l) at the time of the call. The pair (P, P') is said to be consistent with L if the two constraints on the expansion factor and the angle of rotation are satisfied for the list $L \bigcup \{(P, P')\}$.

5 The Complexity of the Algorithm

Let $P(n)$ be the complexity of the procedure *Search*, where n is the number of points in the boundary C, we have:

$$P(n) = \alpha + pM(P(n-1) + \beta) + (1-p)$$

where α is a constant accounting for the first **IF** statement, p the probability of success of the condition **IF** *Consistent*, and β a constant accounting for the **IF** *Improvement* statement. As stated earlier, M is the number of potential pairings tested for each point.

Among the M potential pairings, only one[2] satisfies the consistency test. Hence $p = 1/M$. The expression of $P(n)$ can be rewritten as:

$$P(n) = \alpha + P(n-1) + \beta + 1 - \frac{1}{M}$$

If we put the constant $\gamma = \alpha + \beta + 1 - \frac{1}{M}$ we have:

$$P(n) = \gamma + P(n-1)$$

By recursively substituting $P(n-1)$ in $P(n)$, we obtain:

$$P(n) = n\gamma$$

Since the procedure *Search* is called M times, it follows that the complexity of the approach is $O(n) = Mn$.

6 Evaluating the Method

The number of pairs obtained is a good indication of the correcteness of computed transformation; if this number is not large enough, it can be easily decided that the parameters of the transform could not be computed using this technique. This is the case for straight lines. In *2-D* recognition, if a data curve and a model curve have enough points of significat curvature, but the number of pairings between them is low, a decision can be taken to reject the curve as part of a model object. In motion context, a low number of pairings implies that the model used for computing motion does not fit the physical changes in the scene. In fact, the ratio

$$r = \frac{p}{n} \tag{14}$$

where p is the number of paired points, and n is the smallest of number of points of the two curves, is a self-verifying indicator of the goodness of the model. The computed transform parameters are taken to be reliable if $r \geq 0.5$, which implies that 50% of the points of significant curvature are correctly matched in accordance with the constraints.

[2]If more than one does, they must be at the same position, and hence are the same.

7 Results

In this section, we present results of the computation of the parameters of the transform. Comparisons are drawn with Ma and Chen's technique, which was actually implemented for this purpose.

Boundary	Method	k	θ	(t_x, t_y)
(a)	HT	1.0	90°	(52,2)
	AC	2.0	0°	(10,10)
	CPM	2.0	0°	(10,10)
(b)	HT	0.99	0°	(6,15)
	AC	1.0	8°	(5,10)
	CPM	1.009	8°	(5,11)
(c)	HT	1.0	−9°	(-2,1)
	AC	2.3	10°	(-5,10)
	CPM	2.27	10°	(-4,10)
(d)	HT	0.98	60°	(19,12)
	AC	2.0	4°	(-5,10)
	CPM	1.98	4°	(-5,10)

Table 1. *The parameters computed by the Hough Transform method (HT) and by our method: Constrained Point Matching (CPM). (AC) are the actual parameters. See* **Figure** *6 for curves (a), (b), (c) and (d).*

Judging from the results displayed in [3], our technique performs at least as well as Cohen *et al.*'s for the particular case of boundaries undergoing *similarity* transformation (uniform deformation). It is also more conservative, for their technique is iterative and is based on a first guess.

Figure 6 (c) and (d) are designed to show that our technique accounts properly for disruptions (indicated by arrows) appearing at end points due to occlusion. The parameters found are the correct parameters which would transform one boundary to the other regardless of their differences at end points. Compare these results with those obtained with [7]'s method (see **Table** 1).

The following set of real experiments was performed as part of a correspondence-based object separation module [5]. Boundaries are first extracted from sequences of images [11], and their correspondence is established [10]. Due to the physical setting (conveyor belt) or to the nature of small movements, boundary motion is assumed to follow the (rotation, scale and translation) transform model, which can be cross-checked *a posteriori*.

Figure 8 shows the velocities obtained for the two successive frames of **Figure** 7. Note that the motion of the longest boundary has not been affected by the over-segmentation which resulted in joining two different boundaries. The number of high curvature points pairings on the top part of the boundary, satisfying the correct expansion $k = 1.0$, is large enough to override any perturbation arising from the lower part.

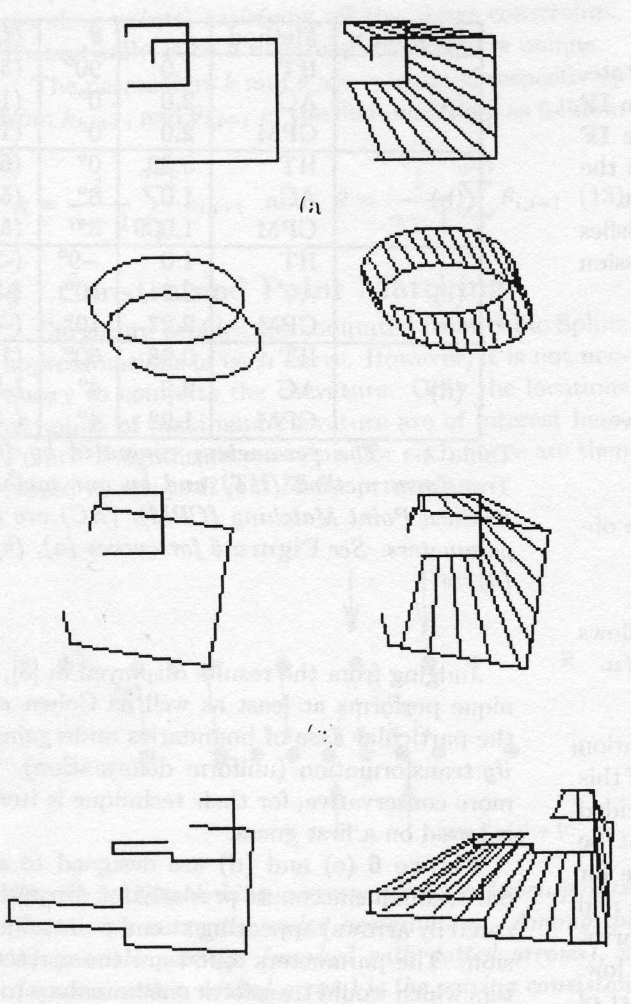

Figure 6. *(a), (b), (c) and (d) are a set of artificial boundaries undergoing controlled transformations. The parameters computed for these transformations are displayed in* **Table** *1, and sketched here by point-to-point mapping along the curves. Experiments (c) and (d) in particular, show the ability of our technique to deal with occlusion disruptions. These disruptions are indicated by arrows.*

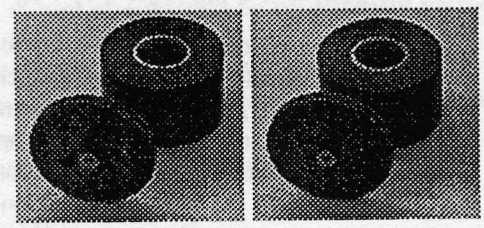

Figure 7. *A real motion sequence in which the object at the back undergoes a translational motion to the left. This motion is assumed to follow the affine transform model.*

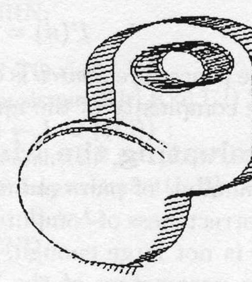

Figure 8. *Mapping obtained for the two-frame sequence of* **Figure** *7.*

Figure 10 shows the velocities obtained for the two successive frames of **Figure** 9.

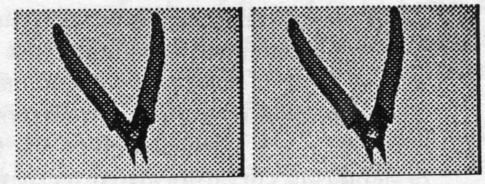

Figure 9. *The image on the right was obtained by moving the camera forward. This movement follows the affine transform model.*

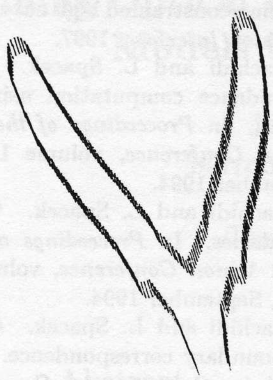

Figure 10. *Mapping obtained for the two-frame sequence of* **Figure** *9, in which the wire snips undergo a small expansion, because of the forward motion of the camera.*

Figure 13 shows the velocities obtained by our approach and the Hough Transform approach for the two successive frames of **Figure** 11.

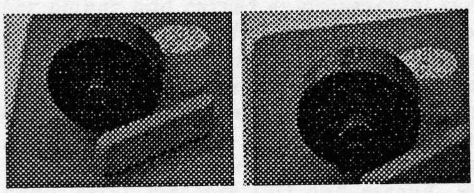

Figure 11. *A simulated conveyor belt sequence: As the mouse pad is manually moved, the different objects undergoing a* 3-D *translation towards the camera. This motion is assumed to follow the Affine transform model.*

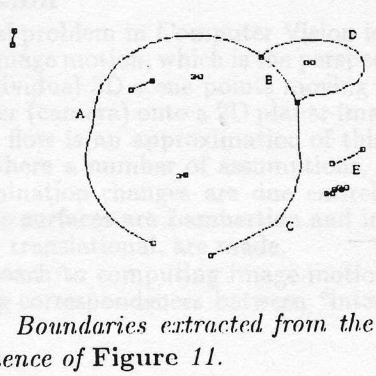

Figure 12. *Boundaries extracted from the left image of the sequence of* **Figure** *11.*

Figure 13. *Left: the motions obtained by our technique. Right: the motions obtained with the Hough Transform approach for the sequence of frames in* **Figure** *11.*

Boundary	Method	k	θ	(t_x, t_y)
Bdy A	HT	0.79	0°	(-73,-77)
	CPM	0.80	0°	(-39,-63)
Bdy B	HT	0.84	−1°	(-24,-48)
	CPM	0.83	0°	(-26,54)
Bdy C	HT	0.01	180°	(-41,11)
	CPM	0.87	0°	(-27,-95)
Bdy D	HT	0.98	41°	(5,-105)
	CPM	0.66	0°	(1,-55)
Bdy E	HT	0.0	0°	(0,0)
	CPM	1.33	0°	(-21,-53)

Table 2. *The parameters computed by the Hough Transform method (HT) and by our method: Constrained Point Matching (CPM). See* **Figure** *12 for labels A, B, C, D and E.*

In this sequence, the objects undergo a translation towards the camera, as a result of the forward motion of the mouse pad (to simulate a conveyor belt). **Table** 2 gives the values of the parameters obtained for the boundaries of the reverse sequence by the two methods. It can be seen clearly that the Hough Transform method *i.e.,* Ma and Chen's, fails to find plausible parameters for boundary C. This is because of the occlusion effect, which our technique was designed to overcome. Also, the Hough Transform technique fails to find the correct parameters for boundary D. This, however, is due to the sub-optimalities in curvature values computed by a crude method. Our technique, on the other hand, delivers correct parameters because curvature values are not used, but rather we use the locations of high curvature points. As expected, boundaries A, B and C all have similar expansion factors (0.80,0.83 and 0.87). Boundary D on the other hand has a different expansion factor 0.66 which actually underlies the difference in depth with boundaries A, B and C. The parameters found for boundary E, however, are erroneous. This is because there is not a sufficient number of high curvature points to capture the motion. Note that Ma and Chen's technique failed too.

Conclusion

The *2-D* motion obtained by our technique has been successfully used to separate objects in the scene [12]. Finally, the various experiments demonstrate the robustness of our approach to determine the parameters of the transform. This robustness stems from:

- the formulation of the parameters of motion, namely, the expansion and the angle of rotation in terms of high curvature points pairings rather than the curvature itself;
- the algorithm developed to establish the matching between points of significant curvature, which integrates the expansion factor and the angle of rotation to constrain the search space. This algorithm has been adapted to compute axis of symmetry of two curves [9].

References

[1] N. Ayache and O. D. Faugeras. Hyper: a new approach for the recognition and positioning of two dimensional objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(1), 1986.

[2] C. H. Chien and J. K. Aggarwal. Model construction and shape recognition from occluding contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(4):372–389, 1989.

[3] I. Cohen, N. Ayache, and P. Sulger. Tracking points on deformable objects using curvature information. In *Second European Conference on Computer Vision*, pages 458–466, 1992.

[4] L. S. Davis. Shape matching using relaxation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1:60–72, 1979.

[5] T. eddine Rachidi. *Separating Solid Objects using the Correspondence of Boundaries*. PhD thesis, Department of Computer Science, University of Essex, Colchester CO4 3SQ, U.K., December 1994.

[6] D. P. Fairney and P. T. Fairney. On the accuracy of point curvature estimators in a discrete environment. *Image and Vision Computing*, 12(5):259–265, 1994.

[7] S. D. Ma and X. Chen. Hough transform using slope and curvature as local properties to detect arbitrary 2-d shapes. *IEEE International Conference on Pattern Recognition*, 1:511–513, 1988.

[8] P. M. Merlin and D. J. Farber. A parallel mechanism for detecting curves in pictures. In *IEEE Transactions on Computers*, pages 96–98, January 1975.

[9] T. Rachidi and A. Amar. Computing axis of symmetry using constrained high curvature points matching. In *Vision Interface*, 1997.

[10] T. Rachidi and L. Spacek. Boundary-based correspondence computation using the topology constraint. In *Proceedings of the 5th British Machine Vision Conference*, volume 1, pages 55–64, York, September 1994.

[11] T. Rachidi and L. Spacek. Constructing coherent boundaries. In *Proceedings of the 5th British Machine Vision Conference*, volume 1, pages 296–304, York, September 1994.

[12] T. Rachidi and L. Spacek. Occlusion analysis using boundary correspondence. In *Proceedings of the 2nd Asian Conference on Computer Vision*, volume 3, pages 432–436, Singapore, December 1995.

[13] T. P. Wallace, O. R. Mitchell, and K. Fukunaga. Three dimensional shape analysis using local shape descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(3):310–323, 1984.

[14] H. Wallach, editor. *On Perception*, chapter On Perceived Identity: 1. The Direction of Motion of Straight Lines. Quadrangle, New York, 1976.

[15] M. Worring and A. W. M. Smeulders. The accuracy and precision of curvature estimation methods. In *IAPR*, volume 3, pages 139–, 1992.