Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

PERGAMON

Pattern Recognition 37 (2004) 525–542

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Viewpoint independent matching of planar curves in 3D space

B.J Liang[a], A.M Wallace[b,*]

[a] *Department of Computer Science, University of York, York YO10 5DD, UK*
[b] *School of Engineering and Physical Sciences, Heriot-Watt University, Edinburgh EH14 4AS, UK*

Received 13 January 2003; accepted 22 August 2003

Abstract

We present a new approach to match planar curves using the weak perspective projection model. This is based on a set of shape parameters that can be extracted from a closed or open contour, derived from the original image as a $\theta(s)$ boundary code. In order to reduce the complexity and increase the robustness of the matching process, the original parameters are reduced to a set of three intermediate variables, each of which can be calculated independently. These variables are contained within a system of linear equations which define the angles and the ratio of the heights of corresponding point pairs on the two contours with respect to a floating coordinate system. The shape matching process is scale and orientation independent, and the original parameters that describe the relative pose of the two contours in 3D space can be recovered subsequently. The approach can be applied to "featured" and "featureless" contours, to whole and partial contours, and is demonstrated on images of contours and mechanical parts and tools to recover identity and pose.
© 2003 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.

Keywords: Planar curve matching; Weak perspective projection; Scale and orientation independence

1. Introduction

Planar curves provide a rich source of information about a scene. For example, many objects have planar or near-planar features on their surfaces, and the outline of a 3D object can often be treated as a planar curve. These curves are usually, but not necessarily, formed from linked edges extracted from an intensity image. "Virtual" planar curves can also be formed by linking feature points, for example to form the skeleton or medial axis transform of an object. In contrast to surface descriptors such as colour, shading and texture, contours are not directly dependent on illumination parameters.

A 2D curve descriptor is a mathematical form to describe a planar curve, information preserving if the original curve can be recovered, otherwise an information non-preserving descriptor. Chain codes are information preserving and have been employed extensively, recording the orientation of list of edge elements along a curve, either in discrete, e.g. Refs. [1,2], or continuous $\theta(s)$, e.g. Ref. [3], form. Discrete 4- or 8-connected chain codes are dependent on the orientation and scaling of the curve, but, provided a robust code can be extracted, $\theta(s)$ descriptors can be used to match contours rotated and translated within or slightly out of a plane [3]. To match chain codes of planar curves oriented arbitrarily in 3D space, it becomes necessary to associate their descriptors with projection rules. For example, the arc-height function [4] has similarities to the basic chain code, moving a fixed length chord along a curve; the height from the center of the chord to the curve is used to parameterise the curve.

The Hough transform [5,6] has been applied extensively to represent and match full or partial contours, transforming the constituent points into curves or surfaces in parameter space, or into an R-table in generalized form. It is robust but information non-preserving; for example the extent of the contour in 2D or 3D space is lost. If arbitrarily viewed contours in 3D are transformed into a 2D transform space without account of the projection, then the transforms do not provide an indication of the correct match. The Fourier

* Corresponding author. Tel.: +44-131-451-3423; fax: +44-131-451-4155.
E-mail addresses: bojian@cs.york.ac.uk (B.J Liang), a.m.wallace@hw.ac.uk (A.M Wallace).

0031-3203/$30.00 © 2003 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2003.08.015

descriptor is another transformation that is invariant to rotation and scaling in a single 2D plane, but not in 3D. They have been used for the classification of partial 2D shapes [7] and for the recognition of 3D objects by their silhouettes [9], but are highly sensitive to errors in the curve length. Other authors (e.g. Ref. [8]) have used contours as primitive groups to infer surfaces to compare against a volumetric rather than a contour model, but such approaches have their own difficulties and fall outwith the scope of this paper.

Approaches based on invariants, reviewed in Refs. [10,11], are attractive as they are by their nature invariant to scale and projection as appropriate, and can be used to constrain otherwise complex feature matching [12]. However, they do not provide geometric information on scaling and position, and it may not always be possible to provide a unique classification as one set of invariants may correspond to a class of curves. For example, in identifying planar curves in 3D space, Gyganski et al. [13] developed an affine invariant parameterization using estimates of the first- and second-order derivatives of the sampled image data. Orthonormalization of the parametric curve functions was used, and correlation of the functions with each library image gave a peak in response. Roh and Kweon [14] also employed a 2D descriptor based on affine invariants of five equally spaced co-planar points on a contour, and a hash table to retrieve potential matches but, like many previous approaches, did not deal with open curves.

Mokhtarian and co-authors [16] have employed for several years a multi-scale, curvature-based shape representation technique for planar curves, formed by convolving the normalized arc length parameter with a Gaussian filter of variable width σ. To overcome the shortcomings of such a representation, two developments were also proposed, the renormalized and the re-sampled curvature scale space image. These representations have been tested on large databases, primarily subject to affine transformation in a 2D plane, but some preliminary work has been done on 3D transformations [15]. Snakes [17] are energy minimizing splines pulled to coincide with the image contours by image forces but held by predefined external constraint. Like any optimization problem, this depends on initialisation adjacent to a clearly defined minimum, or alternatively some element of random search is necessary. To establish an active snake model, a global optimization technique has been applied to a multi-level Markov Random Field (MRF) energy environment [18] but the procedure is not fully autonomous and does not handle pose variation.

In summary, shape matching of contours and the recovery of the geometric parameters are difficult problems in the most general sense, and the published work to date, of which the above is just a fraction, rely on disparate constraints of varying strength. In the context of this paper, the most notable omission in many of these techniques is the inability to handle arbitrary projection of the 2D contour in 3D space. Here, we consider rigid (i.e. not flexible) contours, but the shape matching process is scale and orientation independent and the original geometric parameters can be recovered. The contours may be wholly visible or partially occluded and may or may not include significant features along their length.

2. Parameterizing the planar curve

We use a $\theta(s)$ code to represent contours formed from linked edge data [19]. A continuous 2D curve can be described by the functions

$$x = x_0 + \int_0^l \cos[\theta(s) + \theta_0]\,\mathrm{d}s,$$

$$y = y_0 + \int_0^l \sin[\theta(s) + \theta_0]\,\mathrm{d}s, \tag{1}$$

where (x, y) are the curve coordinates and θ is the inclination of the tangent at distance s along the curve of length l. (x_0, y_0) denotes the starting position. The $\theta(s)$ code has several useful properties summarized below.

- *Translation*. Translation of a curve by (x_t, y_t) is equivalent to a change of the start point (x_0, y_0) of Eq. (1) to $(x_0 + x_t, y_0 + y_t)$.
- *Rotation*. Rotating a curve around its start point by θ_d is equivalent to changing the inclination of the tangent of the start point to $\theta_0 + \theta_d$.
- *Scaling*. Scaling the size of the curve by a factor of k without changing the start point is equivalent to multiplying the integral of Eq. (1) by k,

$$x = x_0 + k\int_0^l \cos[\theta(s) + \theta_0]\,\mathrm{d}s,$$

$$y = y_0 + k\int_0^l \sin[\theta(s) + \theta_0]\,\mathrm{d}s. \tag{2}$$

- *Start point*. When the start point of a curve represented by Eq. (1) (x_0, y_0), is moved along the curve by length L_n to (x_{02}, y_{02}), the new equation can be written as

$$x = x_{02} + L\int_0^u \cos[\theta_2(uL) + \theta_{02}]\,\mathrm{d}u,$$

$$y = y_{02} + L\int_0^u \sin[\theta_2(uL) + \theta_{02}]\,\mathrm{d}u, \tag{3}$$

where

$$x_{02} = x_0 + L\int_0^{L_n/L} \cos[\theta(uL) + \theta_0]\,\mathrm{d}u,$$

$$y_{02} = y_0 + L\int_0^{L_n/L} \sin(\theta(uL) + \theta_0)\,\mathrm{d}u,$$

$$\theta_{02} = \theta(Ln) + \theta_0,$$

$$\theta_2(uL) = \theta[(uL + L_n)|_{Mod(L)}] + \theta(L_n).$$

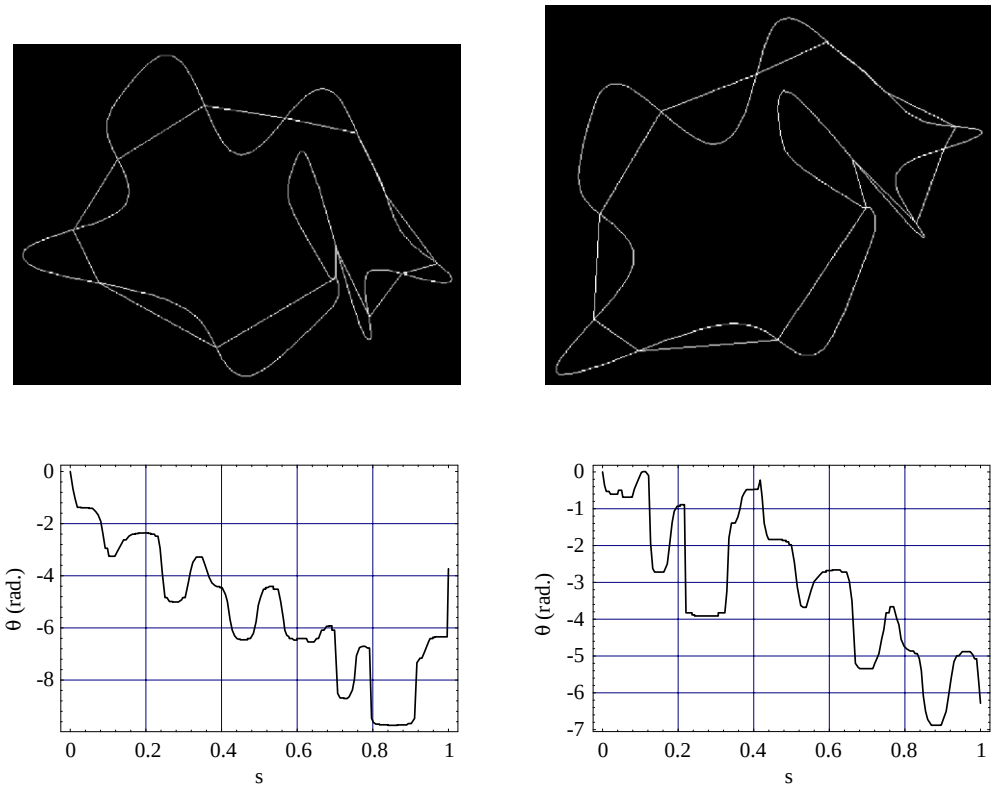

Fig. 1. An example of two projections of a contour, the smoothed $\theta(s)$ curves and the computed inflection points. (a) Inflections on smoothed contour viewed normal to the projection plane. (b) inflections on smoothed contour rotated and viewed at 45° to the projection plane, (c) smoothed $\theta(s)$ curve for (a), (d) smoothed $\theta(s)$ curve for (b).

- *Direction of encoding*. Changing the direction of encoding along the curve is equivalent to changing the $\theta(l)$ function to $-\theta(L-l)$, where L is the total length of the curve.

In principle, peaks in the first differential of the $\theta(s)$ code correspond to curvature points that can be used as features in a curve matching strategy. We can also use points of inflection, i.e. where $\theta(s)$ changes from a monotonic increase to a decrease or vice versa, characterised by a change of sign in $\theta'(s)$. Like other authors [20,21], we have employed an iterative smoothing algorithm [22]. This traces the contour in discrete "chain-link" form to remove short segments defined by several rapid changes of sign in $\theta'(s)$. The inflection points shown in the example of Fig. 1 are more easily extracted, but any smoothing process has the potential to reject very fine detail. Indeed, in spite of smoothing techniques, curvature and inflection points are difficult to define due to statistical variation in computing the angle of the contour from image/edge data. Therefore, we describe an approach that can use feature points when these are reliable, but can also cope with occluded and "featureless" contours.

3. Projection of a planar curve

In general, the projected shape of a planar contour onto the image plane is dependent on the viewing parameters in 3D space. If the size of the object is much less than the distance from the camera, it is possible to use weak perspective projection [23,24],

$$x_c = \frac{f}{\bar{Z}}\, x_w,$$

$$y_c = \frac{f}{\bar{Z}}\, y_w, \tag{4}$$

where x_c and y_c are the camera centred image coordinates and $\bar{Z}$ is the mean of the minimum and maximum distance to the object. In Fig. 2, the plane curve c exists in object plane P_2, and intersects the z axis at z_0. The plane P_1 is parallel to the image plane and passes through point $(0, 0, z_0)$ intersecting P_2 at line **A**. **A** is the projection principal direction about plane P_2. Rotating the XYZ coordinate system around the z-axis by angle, $(90-\zeta)$, and the 2D coordinate system on P_2 so that the x axes of both systems coincide with **A**,

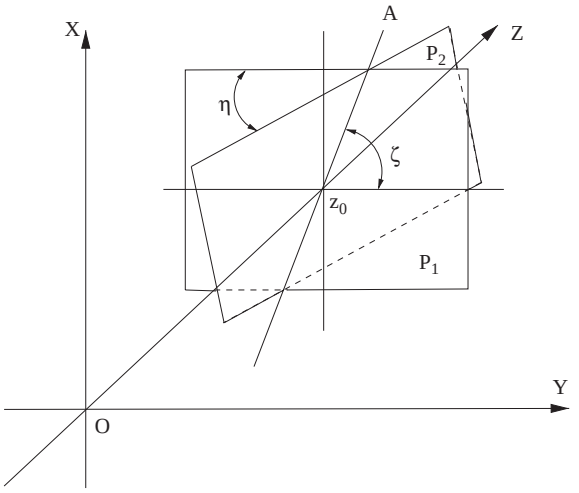

Fig. 2. Representation of a planar contour with respect to the world coordinate system.

we have,

$$\begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} = \begin{bmatrix} x_m \\ y_m \cos\eta \\ z_0 - y_1 \sin\eta \end{bmatrix}, \tag{5}$$

where x_p, y_p, z_p are the 3D coordinates and x_m, y_m are the 2D coordinates on the object plane P_2 after rotation.

4. The Angle-Height (AH) method to match two planar curves

4.1. Shape matching and the floating coordinate systems

In Eq. (1) (x_0, y_0) define the position and are not used to match the shape of the curve. Without loss of generality, x_0 and y_0 can be set to zero. Then, Eq. (1) can be written as

$$x = \int_0^l \cos[\theta(s) + \theta_0]\,\mathrm{d}s,$$

$$y = \int_0^l \sin[\theta(s) + \theta_0]\,\mathrm{d}s. \tag{6}$$

From Eq. (5), a point on the object curve c_p can be written as

$$\begin{bmatrix} x_p(s_p) \\ y_p(s_p) \end{bmatrix} = k \begin{bmatrix} x_m(s_m) \\ \cos\eta\, y_m(s_m) \end{bmatrix}. \tag{7}$$

We express the points, m_m and p_p, on the model, c_m, and object (projected), c_p, curves in parametric form as $(x_m(s_m), y_m(s_m))$, and $(x_p(s_p), y_p(s_p))$, respectively, where s_m and s_p are the curve lengths from the start point of the respective curves. k is a scaling factor, defined as the ratio of the distances of two pairs of corresponding points on the model image curve and the object image curve at the projection principal direction. Hence, the ratio of the tangents of the two polar angles formed by two corresponding points is a constant, $\cos\eta$. From Eq. (7), this can be expressed as

$$\frac{y_p(s_p)}{x_p(s_p)} = \cos\eta\,\frac{y_m(s_m)}{x_m(s_m)}. \tag{8}$$

In practice, the projection principal direction is not known in advance. With reference to Fig. 3, we can define the relationship between the corresponding angles on the object and model curves with respect to axis A as follows:

$$\tan[\theta_p(s_p) + \theta_{p0}] = \cos\eta\,\tan[\theta_m(s_m) + \theta_{m0}]. \tag{9}$$

Axis X_m projects to axis X_p, O_m and m_m project to O_p and p_p, respectively. A_{0m} and A_{0p} are the angles of the two axes X_m and X_p with respect to the axis A, respectively. We call X_m and X_p the *base lines* of the two curves for the respective floating coordinate systems. All local measurements of angle and distance are referred to these base lines. Eq. (9) shows that the ratio of the tangents of the two corresponding angles with respect to the projection principal direction is constant.

To match contours we wish to find η, k, θ_{p0} and θ_{m0} when $c_m = \{x_m(s_m), y_m(s_m)\}$ and $c_p = \{x_p(s_p), y_p(s_p)\}$ are given. To find the solution of Eq. (7), we first separate the variables, and then establish the equations which associate these variables with some intermediate variables.

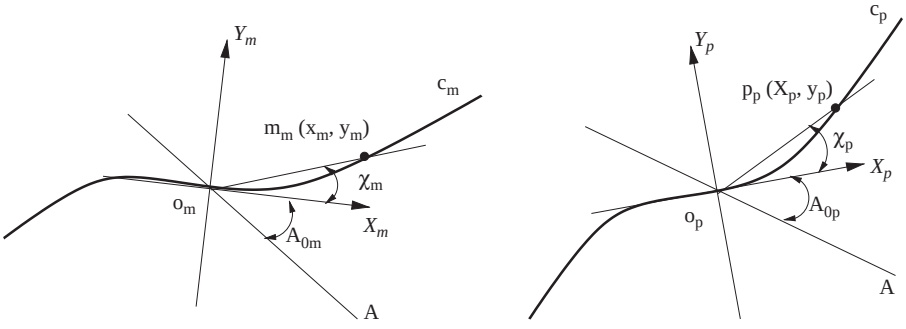

Fig. 3. The floating coordinate systems of the model and object curves.

4.2. Recovery of the intermediate variables

First, the four parameters, η, k, θ_{p0} and θ_{m0}, can be reduced to three. For the moment, assume the start points, s_{pa} and s_{ma}, of the two curves, $\theta_p(s_p)$ and $\theta_m(s_m)$, are matched. Then, we rewrite Eq. (9) as

$$\tan[\theta_p(s_p) + A_{0p}] = \cos\eta \tan[\theta_m(s_m) + A_{0m}]. \tag{10}$$

As $\theta_p(0) = 0$ and $\theta_m(0) = 0$, then

$$\cos\eta = \frac{\tan(A_{0p})}{\tan(A_{0m})}. \tag{11}$$

A_{0p} and A_{0m} represent one pair of corresponding angles measured with respect to the projection principal direction. Therefore, we need to recover k, A_{0p} and A_{0m}. Once we have two pairs of corresponding angles, χ_m and χ_p, that are associated with the base line and the scaling factor k, we are able to recover all the other parameters. Thus, as shown in Fig. 3, the following pair of equations can be applied to any pair of corresponding pairs of points on the two curves to be matched:

$$k = \frac{\int_0^{s_p} \cos[\theta_p(s_p) + A_{0p}]\,\mathrm{d}s_p}{\int_0^{s_m} \cos[\theta_m(s_m) + A_{0m}]\,\mathrm{d}s_m}, \tag{12}$$

$$\frac{\tan(A_{0p})}{\tan(A_{0m})} = \frac{\tan(\chi_p + A_{0p})}{\tan(\chi_m + A_{0m})}. \tag{13}$$

Assuming two pairs of corresponding angles are (χ_{m1}, χ_{p1}) and (χ_{m2}, χ_{p2}). Then, from Eq. (13), A_{0p} and A_{0m} are given by

$$\tan(A_{0m} - A_{0p}) = \frac{\cot\chi_{m2}\cot\chi_{p1} - \cot\chi_{m1}\cot\chi_{p2}}{(\cot\chi_{m2} - \cot\chi_{m1}) + (\cot\chi_{p2} - \cot\chi_{p1})}, \tag{14}$$

$$\tan(A_{0m} + A_{0p}) = \frac{\cot\chi_{m2}\cot\chi_{p1} - \cot\chi_{m1}\cot\chi_{p2}}{(\cot\chi_{m2} - \cot\chi_{m1}) - (\cot\chi_{p2} - \cot\chi_{p1})}. \tag{15}$$

The two solutions should have the same sign. If $\|A_{0m}\| < \|A_{0p}\|$, the solution of the above equations should be modified according to the following condition:

If $A_{0m} > 0$ then $A_{0m} = A_{0m} - \frac{\pi}{2}$, $A_{0p} = A_{0p} - \frac{\pi}{2}$,
else $A_{0m} = A_{0m} + \frac{\pi}{2}$, $A_{0p} = A_{0p} + \frac{\pi}{2}$.

The scaling factor k is dependent on A_{0m} and A_{0p} and can be rewritten as

$$k = \frac{\sin A_{0m} \int_0^{s_p} \sin\theta_p(s_p)\,\mathrm{d}s_p}{\sin A_{0p} \int_0^{s_m} \sin\theta_m(s_m)\,\mathrm{d}s_m}. \tag{16}$$

$\sin A_{0m}, \sin A_{0p}$ are constant, and the integrals are the y coordinates of one pair of corresponding points using the floating coordinate systems.

The ratio, r_y, of the y coordinates of the corresponding points on the two matched curves is constant,

$$r_y = \frac{\int_0^{s_p} \sin\theta_p(s_p)\,\mathrm{d}s_p}{\int_0^{s_m} \sin\theta_m(s_m)\,\mathrm{d}s_m} = \frac{y_p(s_p)}{y_m(s_m)}. \tag{17}$$

Eq. (17) does not involve A_{0p} and A_{0m} and provides a way to project from a model to a object point without reference to the unknown scaling factor k.

$$y_p(s_{pi}) = \frac{y_p(s_{pj})}{y_m(s_{mj})}\, y_m(s_{mi}). \tag{18}$$

Eq. (17) is the *scale matching function* and is the first necessary condition to match two curves. r_y is a necessary intermediate variable.

Applying Eq. (11) to χ_{mi} and χ_{pi} then converting it to cotangent form, we have

$$\frac{\cot A_{0m}}{\cot A_{0p}} = \frac{\cot(A_{0m} + \chi_{mi})}{\cot(A_{0p} + \chi_{pi})}. \tag{19}$$

Let $u = \sin 2A_{0m}/\sin 2A_{0p}$ and $v = (\cos 2A_{0m} - \cos 2A_{0p}/\sin 2A_{0p})$. Then Eq. (19) can be written as

$$\cot\chi_{pi} = u\cot\chi_{mi} + v \tag{20}$$

and

$$\tan(A_{0p} - A_{0m}) = \frac{v}{1+u},$$
$$\tan(A_{0p} + A_{0m}) = \frac{v}{1-u}. \tag{21}$$

Eq. (20) is the second necessary condition that relates the corresponding angles of the two coordinate systems. We call it the *angle matching function*. Hence,

$$u = \frac{\cot\chi_{p2} - \cot\chi_{p1}}{\cot\chi_{m2} - \cot\chi_{m1}}, \tag{22}$$

$$v = \frac{\cot\chi_{p1}\cot\chi_{m2} - \cot\chi_{p2}\cot\chi_{m1}}{\cot\chi_{m2} - \cot\chi_{m1}}. \tag{23}$$

This allows us to compute A_{0m} and A_{0p} from Eq. (21).

The scale matching function and the angle matching function can also be written in the form of an affine matrix:

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} r_y u & r_y v \\ 0 & r_y \end{bmatrix} \begin{bmatrix} x_m \\ y_m \end{bmatrix}. \tag{24}$$

In summary, u, v and r_y are the three intermediate variables, each of which can be calculated independently. The four geometric parameters, η, k, θ_{p0} and θ_{m0}, may be recovered subsequently. As u and v are functions of the angle between the base line and origin-point lines, and r_y refers to the height of the point above the base line, we call this approach the *Angle-Height (AH) method*.

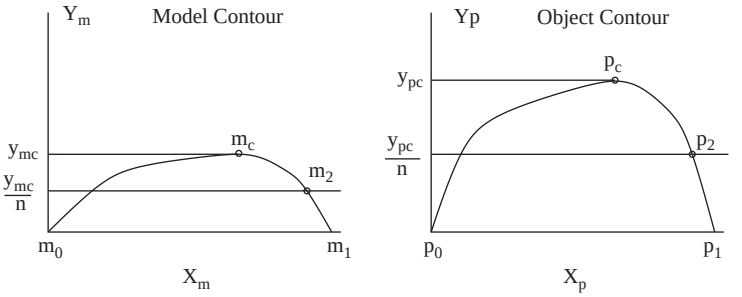

Fig. 4. Identifying the third point to compute the projection.

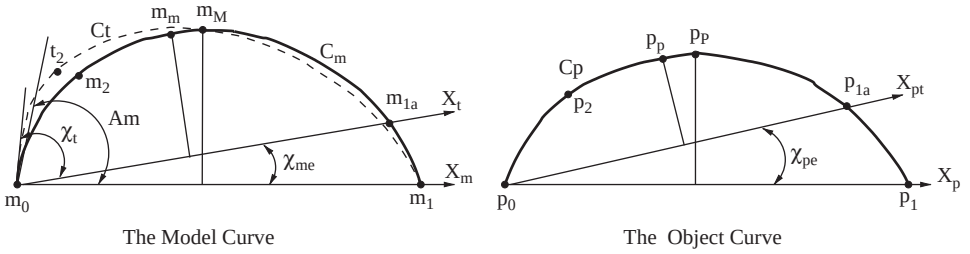

Fig. 5. Defining the projection between the object and model curves.

4.3. Case 1: Two or three pairs of corresponding points are known

To solve Eqs. (17) and (20), we need to find at least one pair of corresponding lines as base lines, two pairs of corresponding angles and one corresponding point pair that is not on the base line. For example, we can use two corresponding point pairs on the curves and their tangents, or three corresponding point pairs. In practice, these could be defined by curvature or inflection of the $\theta(s)$ curve. Using three point pairs so defined, the mapping from object to model curve is fixed and the three intermediate parameters, u, v, and r_y are recovered directly. In many cases, feature points may be unavailable, or it may be desirable to avoid their use, as features based on differential properties may be unreliable due to noise or quantization errors. Further, the initial projected match may not be optimum over the whole curve, requiring refinement, as described in Section 4.6. Hence we now show how the method can be applied with two, one or no initial, corresponding point pairs.

If two pairs of corresponding points are known, m_0 matching p_0 and m_1 matching p_1, we use the lines passing through them as the base lines, as shown in Fig. 4. To recover the three parameters, u, v, and r_y, an additional corresponding point pair must be defined. We could select the *critical points*, m_c and p_c, with the maximum y coordinates as suitable candidates, but as the curvature near these critical points is often small, precise estimation of the x coordinates of the points can become difficult. This problem can be overcome by defining on each curve an additional point, m_2 and p_2,

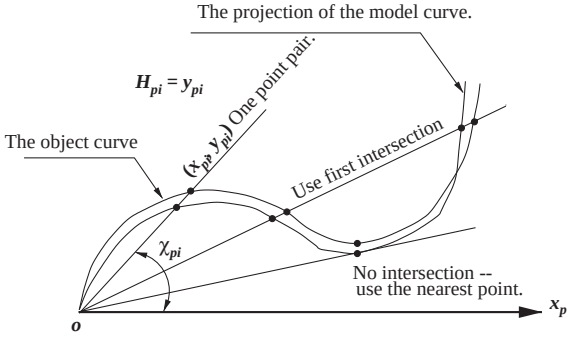

Fig. 6. The three cases for the error sign, plotted as a function of χ_{me}.

such that $y_{m2} = y_{mc}/n$ and $y_{p2} = y_{pc}/n$, where n is a constant. We can define this as a *proportional point*. Once the third point pair is defined, the procedure for recovery of u, v, and r_y is as before.

4.4. Case 2: One corresponding point pair is known

Referring to Fig. 5, the known corresponding point pair is (m_0, p_0). Given another known point, m_{1a}, on the model curve, an arbitrary point on the object curve is selected, say p_{1a}, to establish a base line. Then the third point pair is established by use of the critical (as shown in Fig, 5) or proportional points.

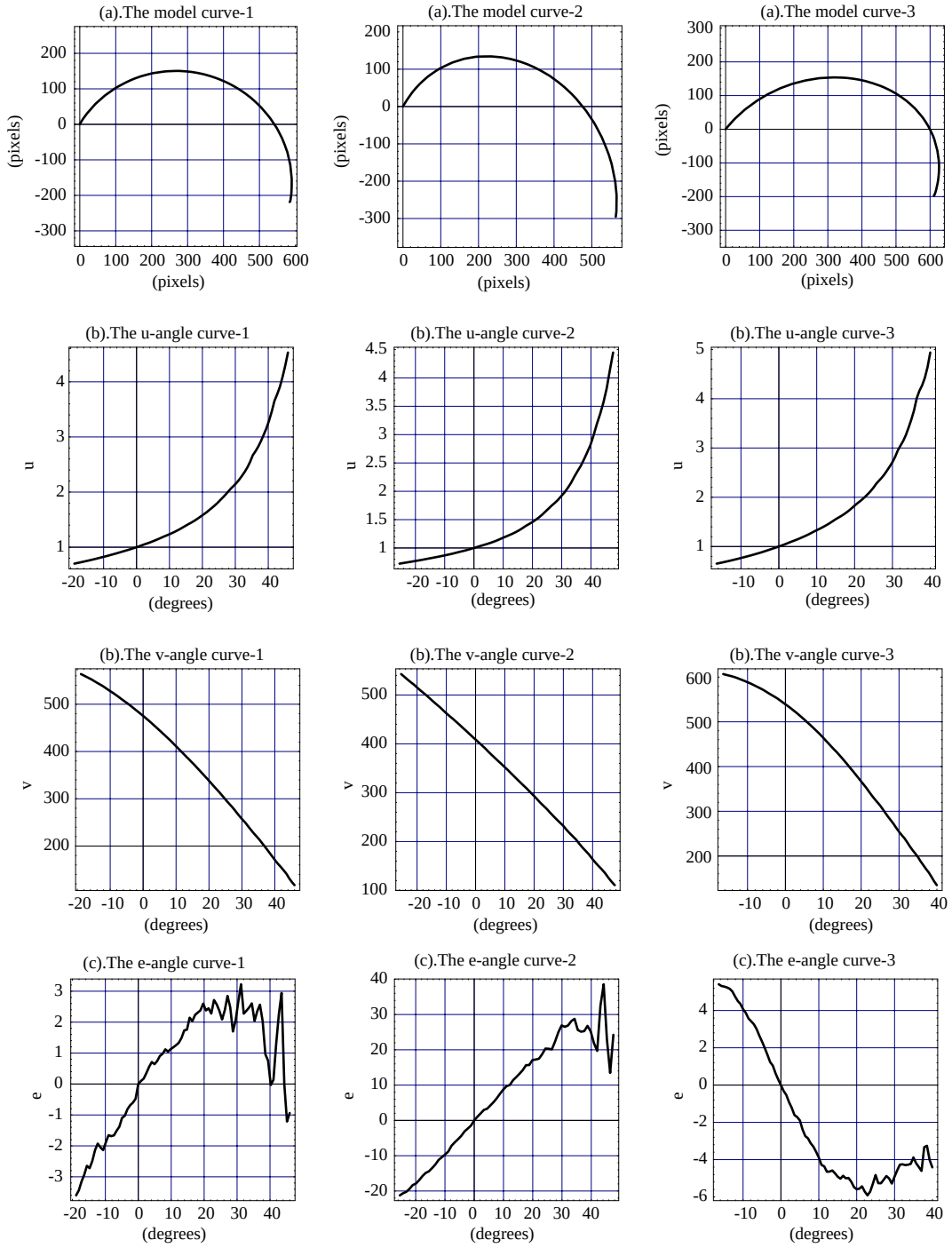

Fig. 7. Assessment and refinement of the match between point pairs.

For the correct projection, $\mathscr{A}$, p_{1a} corresponds to m_{1a}. However, if p_{1a} is not the point corresponding to m_1, the projection matrix recovered is incorrect. This "false" matrix $\mathscr{B}$ does not project the whole model curve onto the object curve. Instead, $\mathscr{B}^{-1}$ projects the object curve to a testing curve, C_t (dashed), which is different from the model curve.

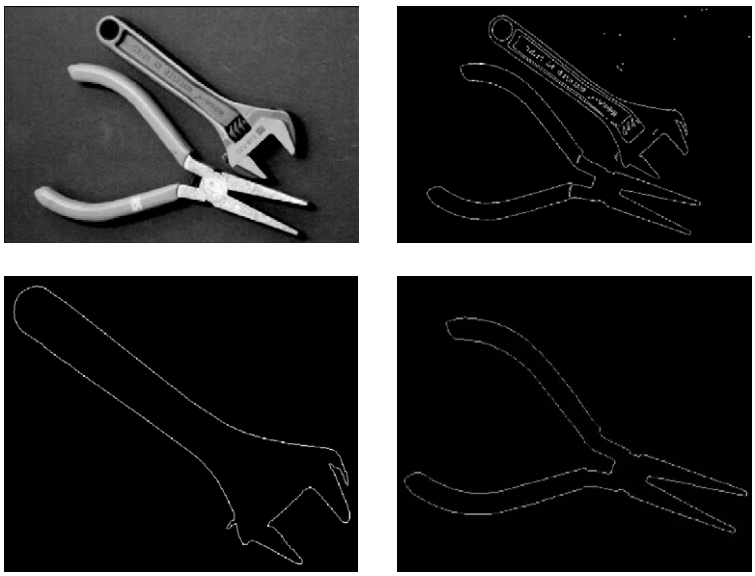

Fig. 8. Modeling the spanner and the pliers: (a) image of separated spanner and pliers, (b) edge image of (a), (c) model curve of the spanner, and (d) model curve of the pliers.

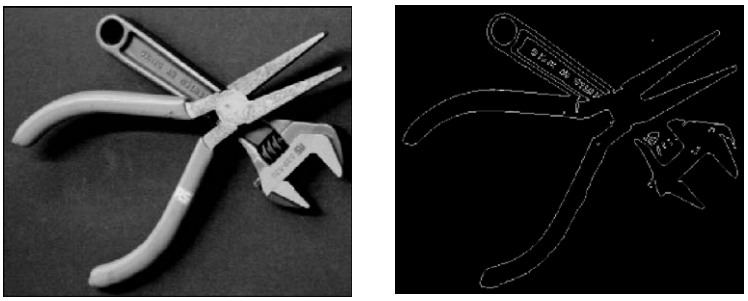

Fig. 9. Image of overlapping pliers and spanner: (a) test image and (b) contours extracted from (a).

The model and the testing curve intersect at at least two points which are on the X_m axis, points m_0 and m_1. C_t also intersects C_m at point m_M. If we can adjust p_{1a} in the right direction according to the relation between the model curve C_m and the testing curve C_t, p_{1a} can eventually be adjusted to coincide with the correct point p_1.

It can be proved that if the object curve is the projection of the model curve, C_t, it is fully defined by the model curve C_m itself. In particular, point p_{1a} projects to the model curve at point m_{1a}:

$$m_{1a} = \mathscr{A}\, p_{1a}. \tag{25}$$

This is also applied to point pairs (m_M, p_P) and (m_m, p_p), where m_M and p_P are the critical points of C_m and C_p. Under the "false" projection $\mathscr{B}^{-1}$, p_p, rather than p_P, projects to m_M. Thus, the behavior of the testing curve C_t projected by the "false" matrix $\mathscr{B}$ is fully described by the known model curve, as shown in Fig. 5. Adjusting p_{1a} to p_1 the behavior of the testing curve C_t is exactly the same as adjusting m_{1a} to m_1—the self-projection property. This is important as we can predict "how wrong the matrix $\mathscr{B}$ is" by studying the projection of the object curve with respect to the model curve.

In Fig. 5, the tangents of the testing curve and the model curve at the origin are χ_t and A_m, respectively. The *error angle* e_a is defined by

$$e_a = A_m - \chi_t - \chi_{me}. \tag{26}$$

The sign of the error angle indicates that the testing curve is on the right or left side of the model curve near the origin of system $X_m - Y_m$. By checking the sign of the error angle, we can adjust the position of p_{1a} on the object curve until the error angle is less than a given tolerance. For example, a simple search rule can be defined; if C_t is to the right of C_m, move p_{1a} down, else, move up.

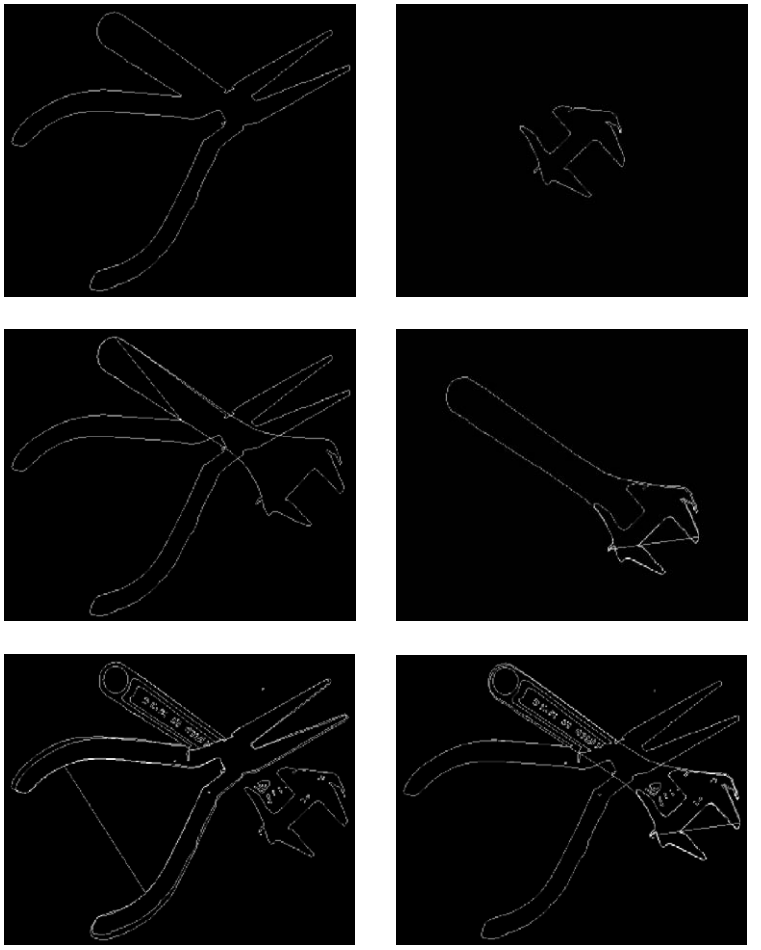

Fig. 10. An example of matching an occluded object: (a) extraction of the first contour for matching, (b) extraction of the second contour for matching, (c) matching the first contour to the pliers, (d) matching the second contour to the spanner, (e) superimposition of the pliers model onto the original contours, and (f) superimposition of the spanner model onto the original contours.

As $\chi_{me} \to 0$, $de_a/d\chi_{me}$ may be larger than, less than or equal to 0. If $\lim_{\chi_{me}\to 0} de_a/d\chi_{me} \neq 0$, a subdivision approach can be applied to find the right point p_1 on the object curve rather than blind search. The first two cases imply that $d\chi_t(0)/d\chi_{me} \neq -1$. In the third case, $d\chi_t(0)/d\chi_{me}(0) = -1$. Hence, we cannot determine the search direction from the sign of the error angle. This case can be avoided by a properly selected point p_1. In fact,

$$\lim_{\chi_{me}\to 0} \tan'(\chi_t + \chi_{me}) = \frac{1 + d\chi_t/d\chi_{me}}{\cos^2 \chi_t}. \tag{27}$$

Let the above equation be equal to 0. Then, we have

$$\frac{d\chi_t}{d\chi_{me}} = -1. \tag{28}$$

As χ_t is non-linear with respect to χ_{me}, there must be other points in which $d\chi_t/d\chi_{me} \neq -1$. From the uniqueness theorem [22], the testing curve can only coincide with the model curve for one value of χ_{me}. The continuous change of the testing curve as point m_{1a} moves along curve C_m ensures that we can find a point for which Eq. (28) is not tenable. As χ_{me} is increased from 0 to t_m monotonically, χ_t changes from 0 to t. The new X_m axis, which defines the new m_1, can be reset between angle 0 and t_m. Then, the error angle will have a different sign when the point m_{1a} is on different sides of point m_1 (which is equivalent to that point p_{1a} being on a different side from p_1). Fig. 6 illustrates the three typical cases for the sign of the error angle. Curve 1 is part of a circle which is a symmetric curve, curve 2 is "sheared" to the left side and curve 3 is "sheared" to the right side. The parameters u, v and the error angle are plotted respectively as functions of angle χ_{me}. (The jitter in the error angle is due to quantization error as the angle becomes larger; the length of curve above the horizontal axis is shorter which leads to

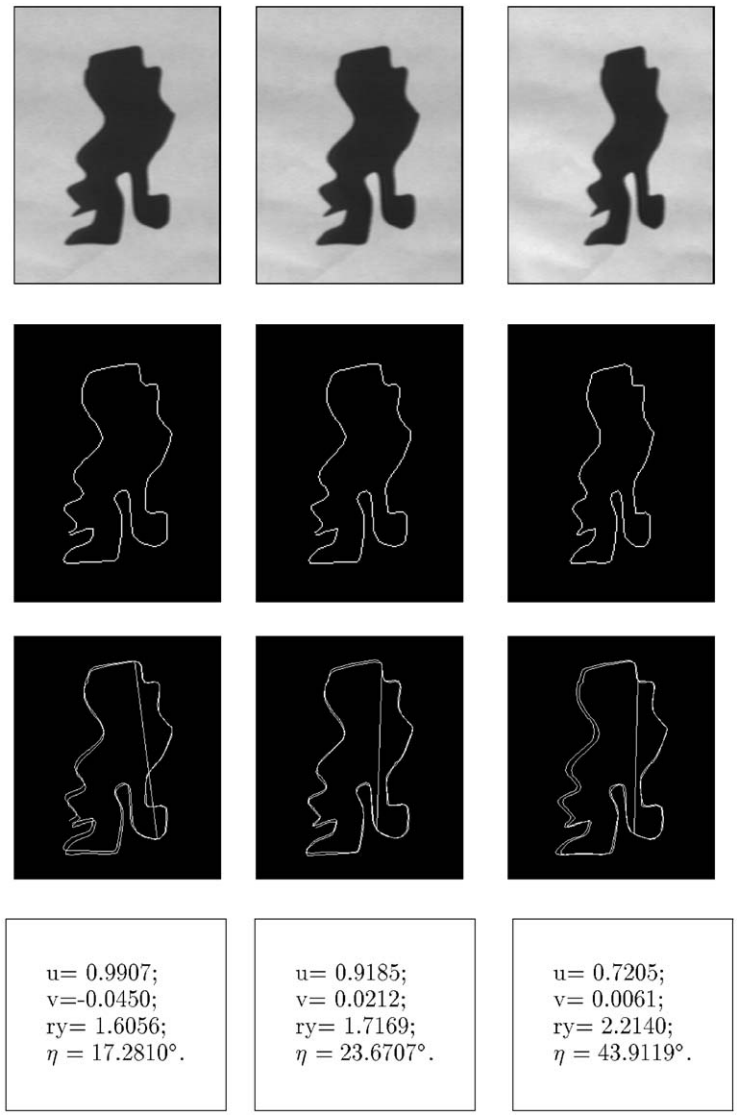

Fig. 11. Matching of object contours to planar contour model: (a) 0° view, (b) 20° view, (c) 40° view, (d) contour of (a), (e) contour of (b), (f) contour of (c), (g) match (d) to model, (h) match (e) to model, (i) match (f) to model, (j) parameters for (d), (k) parameters for (e), and (l) parameters for (f).

a mis-match of the top point.) The sign of the error angle as a function of the sign of angle χ_{me} (also χ_{pe}) is used to guide the search. Alternatively, the sign of the error angle can be replaced by the sign of the error of the testing points, the sign of $e_x = \chi_{m2} - \chi_{t2}$ in Fig. 5. The search terminates when this error angle is sufficiently small.

4.5. Case 3: No corresponding point pairs are known

This is the most general situation for matching two curves and is an extension of the case in Section 4.4. Without any feature information, a single pair of matching points can only be established by exhaustive search. In practice, after an initial point, m_0, is defined on the model curve, each point on the object curve is guessed as the corresponding point. For each possible pair of points, the procedure of the previous section is repeated until the best result is found. This is more complex than the previous cases, because the search is less constrained, but even with no feature data, as the searching is limited to one dimension, the search space is small.

4.6. Assessment and refinement of the estimation

Using the estimated parameters (u, v and r_y), each point on the model curve has an angle $\chi_{mi} = \text{arccot}(x_{mi}/y_{mi})$ that

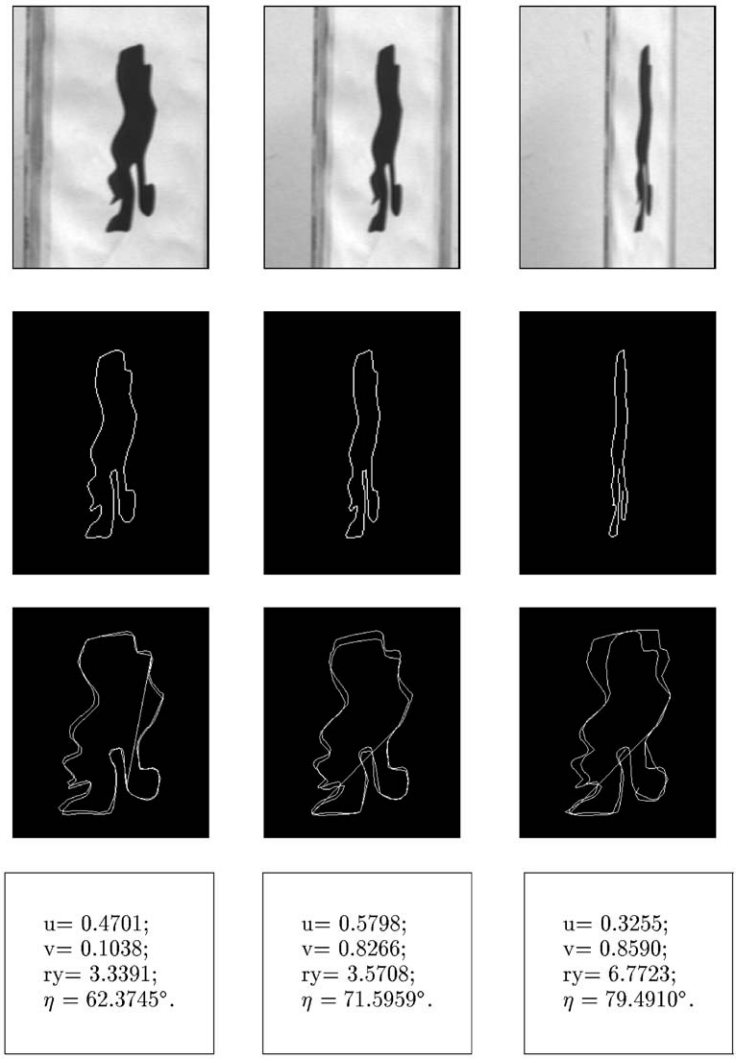

Fig. 12. Matching of object contours to planar contour model: (a) 60° view, (b) 70° view, (c) 80° view, (d) contour of (a), (e) contour of (b), (f) contour of (c), (g) match (d) to model, (h) match (e) to model, (i) match (f) to model, (j) parameters for (d), (k) parameters for (e), and (l) parameters for (f).

can be projected onto the object curve as χ_{pi} by the angle matching function, Eq. (20). As illustrated in Fig. 7, the straight line $y_p = \tan\chi_{pi}\ x_p$ intersects the object curve at point (x_{pi}, y_{pi}) which is estimated to correspond to model point (x_{mi}, y_{mi}). Fig. 6 also shows that the projected line may intersect the object curve more than once or not at all. In the former case, the first intersection is used, in the latter case the point on the curve nearest to the projected line is chosen. For each point, (x_{pi}, y_{pi}), the height $H_{pi} = y_{pi}$ from the object curve is recovered. All these points on the object curve satisfy the angle matching function but may not satisfy the scale matching function. How these points accord with the scaling matching function is used to assess how well the two curves match. Assume there are N point matches between the model and object curves. Each pair is denoted,

$$pp_i((x_{mi}, H_{mi}), (x_{pi}, H_{pi})) \quad (0 \leqslant i < N). \tag{29}$$

We define:

$$e_i = (1 + \|\cot\chi_{pi}\|)H_{pi}(r_y H_{mi} - H_{pi}) \tag{30}$$

as the *single point matching error*. The absolute value, $d_i = \|e_i\|$, is the city-block distance between point p_i and the projection of point m_i. The simple mean, $\bar{d}_s = 1/N \sum_{i-0}^{N-1} d_i$ of d_i is used as one error metric between the model and object curves.

In order to handle partial matches between contours, we can also consider a well-matched curve segment as one in which all the matching errors, d_i, are less than a given

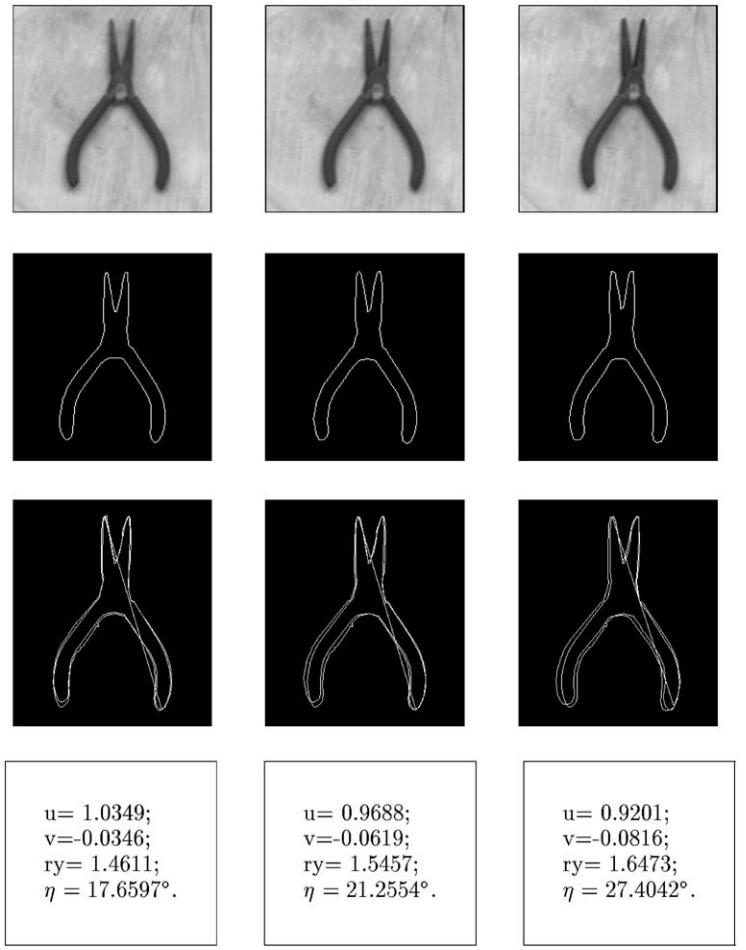

Fig. 13. Matching of object contours to small scale model of normally viewed pliers: (a) 10° view, (b) 20° view, (c) 30° view, (d) contour of (a), (e) contour of (b), (f) Contour of (c), (g) match (d) to model, (h) match (e) to model, (i) match (f) to model, (j) parameters for (d), (k) parameters for (e), and (l) parameters for (f).

threshold. The percentage of well-matched segments on a curve is defined as $p_w = \sum l_{wi}/l_{total}$, where l_{wi} is the curve length of each well-matched segment and l_{total} is the total length of the model curve. This can be computed on either the partial or full curve; the former can handle occlusion and is faster, but yields less precision. An iterative refinement can be applied to further adjust the second point on the base line, i.e. using a small step around the second point on the base line, the parameters u, v and r_y can be recalculated to minimize d_i.

4.7. Extension of the AH method to allow for arbitrary model and object planes

In the previous discussion, the AH method was applied to cases where the model curves were parallel to the image plane. However, there are many cases, e.g. in multi-camera, stereo or video sequence matching where this may not be possible and it is necessary to extend the AH method to work in the more general case.

Assume that $\mathbf{P}_1$ and $\mathbf{P}_2$ are two planar image curves on different planes that are not parallel to the image plane, while $\mathbf{M}$ is a corresponding image curve that is parallel to the image plane. Using the same base line on image curve $\mathbf{M}$, we have

$$\mathbf{M} = \mathscr{A}_1\mathbf{P}_1, \tag{31a}$$

$$\mathbf{M} = \mathscr{A}_2\mathbf{P}_2, \tag{31b}$$

where $\mathscr{A}_1$ and $\mathscr{A}_2$ are affine projection matrices with the upper triangular form. Without using the unknown model curve $\mathbf{M}$, $\mathbf{P}_1$ can be directly mapped to $\mathbf{P}_2$ by

$$\mathbf{P}_1 = \mathscr{A}\mathbf{P}_2 \tag{32}$$

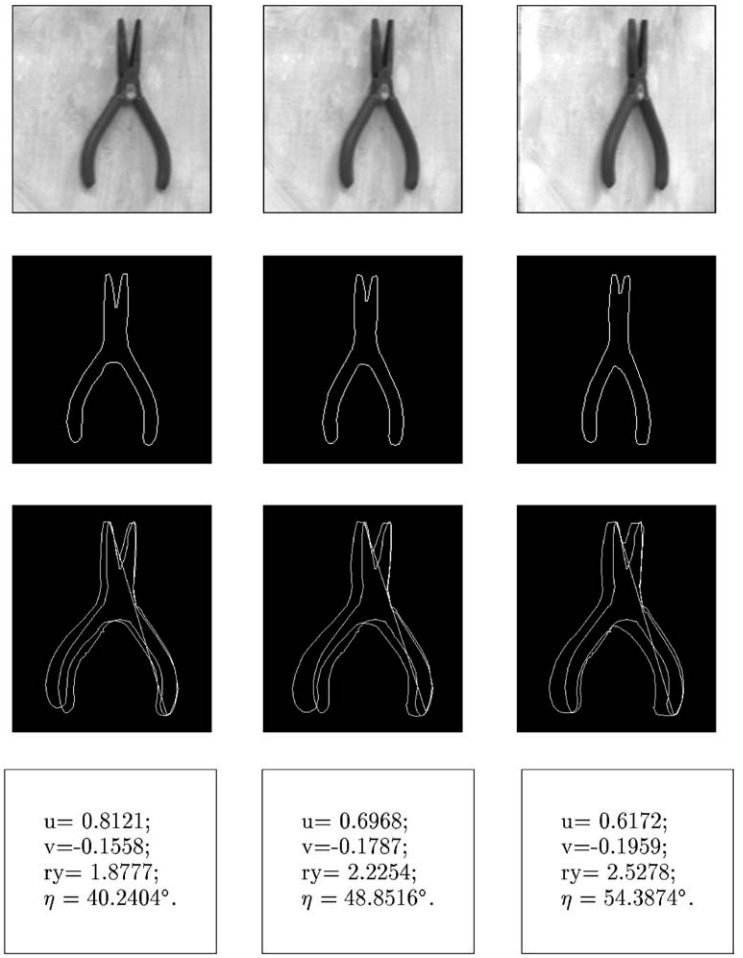

Fig. 14. Matching of object contours to small scale model of normally viewed pliers: (a) 40° view, (b) 50° view, (c) 60° view, (d) contour of (a), (e) contour of (b), (f) contour of (c), (g) match (d) to model, (h) match (e) to model, (i) match (f) to model, (j) parameters for (d), (k) parameters for (e), and (l) parameters for (f).

where $\mathscr{A} = \mathscr{A}_1^{-1}\mathscr{A}_2$. It can be proven that $\mathscr{A}$ is also an upper triangular matrix. This has the same form as the basic equations of the AH method (Section 4.2), hence the same principles can be applied to match two planar curves with arbitrary orientation in 3D space.

5. Experimental evaluation of the AH method

In this section, we show a number of examples of the matching of wholly visible and occluded contours, at different angles of projection and scale, applied to featured and featureless contours. Referring to Fig. 8, the two longest contours are chosen as the model curves and encoded as $\theta(s)$ chains.

A larger scale test image in which the pliers occlude the spanner is shown in Fig. 9. The two possible matching contours derived from Fig. 9(b) are shown in Fig. 10(a) and (b). Following application of the matching using two feature pairs, subsequent assessment and refinement, the best matches between the two test curves and the models are shown in Fig. 10(c) and (d). In these and subsequent figures the straight lines shown are the base lines. The first points at which the straight lines intersect the model and object curves are the origins of the floating coordinate systems. The second points are the second corresponding point pair recovered after the search. Figs. 10(e) and (f) show the superposition of (c) and (d) onto the original contour image, matching the pliers and spanner respectively.

To show the effect of variation of angle of projection, objects were fixed on a rotation frame at a distance about 2.5 m from a camera of focal length 25 mm so that, even at an angle of 80° to the projection axis, the weak-perspective projection assumption is still valid. The first example shows

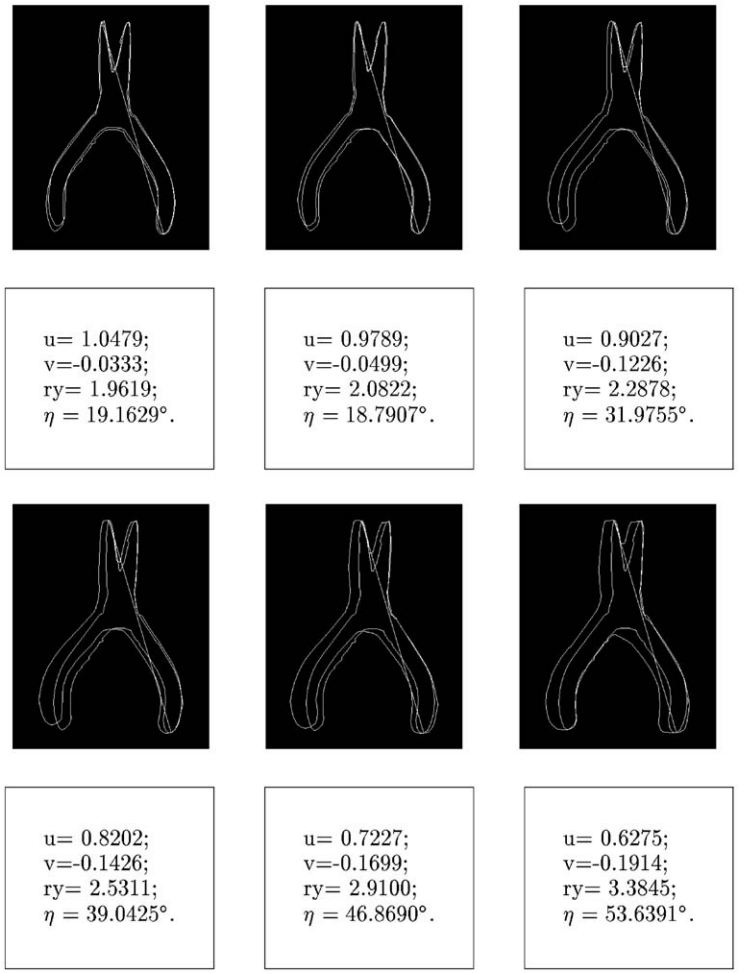

Fig. 15. Matching of image contours in Figs. 13 and 14 to large scale model of normally viewed pliers: (a) Fig. 13(d) to model, (b) Fig. 13(e) to model, (c) Fig. 13(f) to model, (d) parameters of (a), (e) parameters of (b), (f) parameters of (c), (g) Fig. 14(d) to model, (h) Fig. 14(d) to model, (i) Fig. 14(d) to model, (j) parameters of (g), (k) parameters of (h), and (l) parameters of (i).

a succession of images of an irregular contour at different angles to the projection axis. Figs. 11 and 12 show in each column the image of the object, the main contour, the reverse-projection of the target object onto the model and the matching parameters.

Qualitatively, and as measured by $\bar{d}_s$, the matching of model contours to the reverse-projection of the object shows that the scaling factors in the two directions $(r_y u, r_y)$ are good. The angle between the object plane and the image plane is calculated from Eqs. (11) and (21). Large angular errors occur at angles close to 0°. This is because the scaling at the direction perpendicular to the principal direction is equal to the scaling factor at the principal direction multiplied by $\cos\eta$ (Eq. (7)). $\cos(10^\circ) = 0.985$ which means that 1.5 percent of measurement error has the same effect as rotating the object plane by 10°. In addition, the base line does not coincide with the principal direction in this example. The maximum height from the base line to the curve is less than 100 pixels, the quantization error for the digital image is in the range of $\pm\frac{1}{2}$ pixel. These are all in the range of 1.5 percent and make precise recovery of the lower angles difficult. As the angle η becomes larger, the derivative of $\cos\eta$ becomes larger. Hence the values recovered for η are more accurate.

Figs. 13–15 show matching of the pliers, using two normally viewed images at different scales as the model images. The object is not a flat object, hence when the angle is increased, the external contour is not an exact projection of the original contour. Therefore, this example goes beyond the constraints imposed by the theory. At 60° for example, the shape and orientation of the matched contours show significant errors, although at lower angles, the AH approach

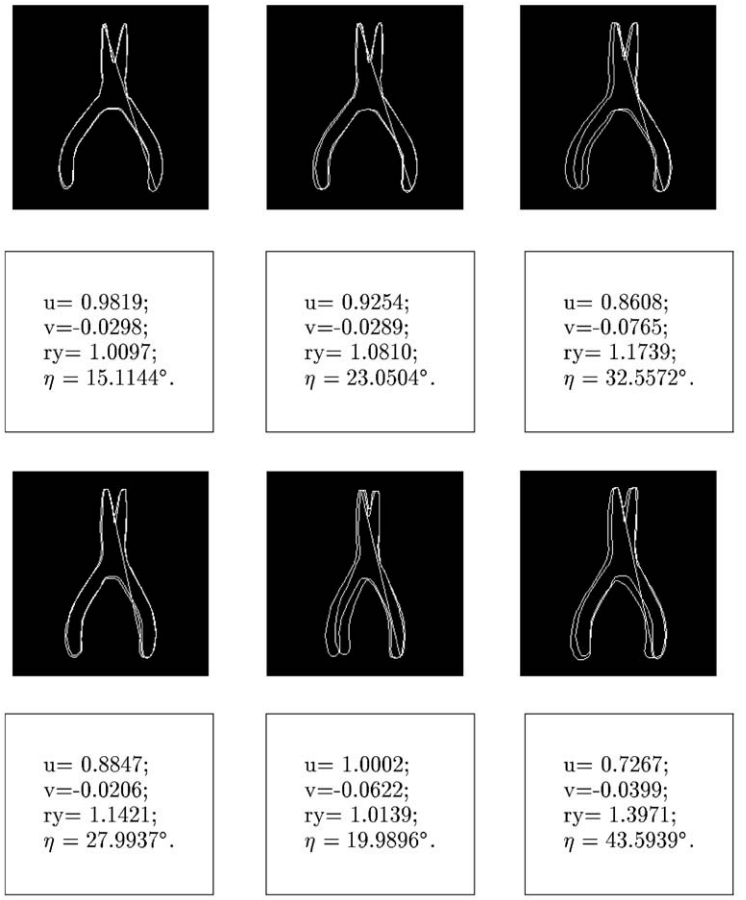

Fig. 16. Matching adjacent images of pliers: (a) 10°–20°, (b) 20°–30°, (c) 30°–40°, (d) parameters of (a), (e) parameters of (b), (f) parameters of (c), (g) 40°–50°, (h) 50°–60°, (i) 40°–60°, (j) parameters of (g), (k) parameters of (h) and parameters of (i).

Fig. 17. Image sequence of a randomly moving key.

works well. This shows that the violation of the planar curve assumption has made it impossible to project from the model to the object curve when the angle between the projection planes is large. In Section 4.7 we argued that there are many cases in which it is more useful to use an arbitrary model and image plane. This is such a case. Rather than use the 0° image as the model plane, one would expect a better result if the previous image of the pliers sequence was used

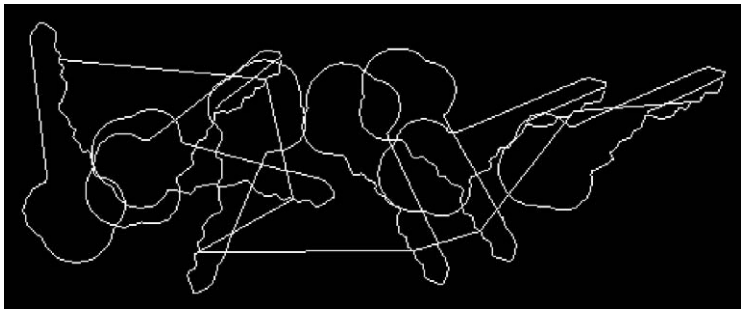

Fig. 18. Tracking the motion of a single point in the key sequence.

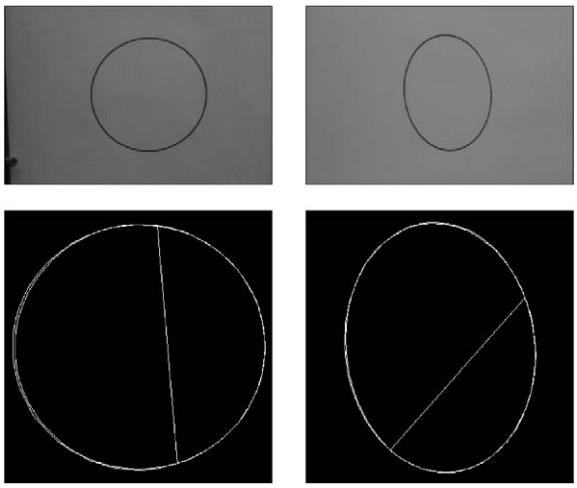

Fig. 19. Matching a circle and an ellipse: (a) circle, (b) ellipse, (c) projection of (a) onto (b), and (d) projection of (b) onto (a).

as a model for the next case. This is shown in Fig. 16. The shape matching is improved but the parameter η shows large errors as the angle between views is small.

A further example shows a key being matched in an eight frame image sequence of which four frames are shown in Fig. 17. The contour of the preceding frame is used to match the main contour of each successive frame. A one-to-one point mapping is established, so that given any point on the contour of the first frame, its corresponding points in the other seven frames can be easily located as "corresponding points chains", defining the motion of a point on the key. One example of a tracked point through all eight frames is shown in Fig. 18.

The next three examples show simple cases in which either no feature data are available, or are not used as they may be considered to be unreliable. In Fig. 19 a circle is matched to its projection, an ellipse at about 45° to the image plane. As any point on the circle may correspond to any point on the ellipse, there is more than one candidate for a possible match. In Fig. 19 the circle is projected on to the ellipse, and vice versa, using the projection parameters recovered by the exhaustive search procedure. The recovered elements of the affine projection matrix by mapping the circle to the ellipse are $a_{11} = 1.2947, a_{12} = -0.4844$, and $a_{22} = 0.9946$, which correspond to an angle between the two planes of 40.039°.

The curve shown in the example of Fig. 20 has several inflection points but these were not used. Instead, an arbitrary point on the model was chosen as the origin of the floating coordinate system. Then a second arbitrary point was chosen at a reasonable distance from the first; the angle formed by the tangents to the curve at the two points was around 20°. Then, the corresponding point pair on the object curve was sought. The two curves are superimposed in Fig. 20(c), showing the result of the assessment and refinement procedure to minimize $\bar{d}_s$. The third example matches a segment of the same contour to a full curve. The search was started from a fixed point on the segment and along the whole curve of Fig. 20(b). The three elements of the affine projection matrix are a_{11}=1.6567, a_{12}=0.7358 and a_{22}=2.0166, which yield $r_y = 2.0166, u = 0.8215$ and $v = -0.3649$. Hence, the angle between the object plane and image plane was 50.12°. This can be seen by the shape shearing of the projection and is

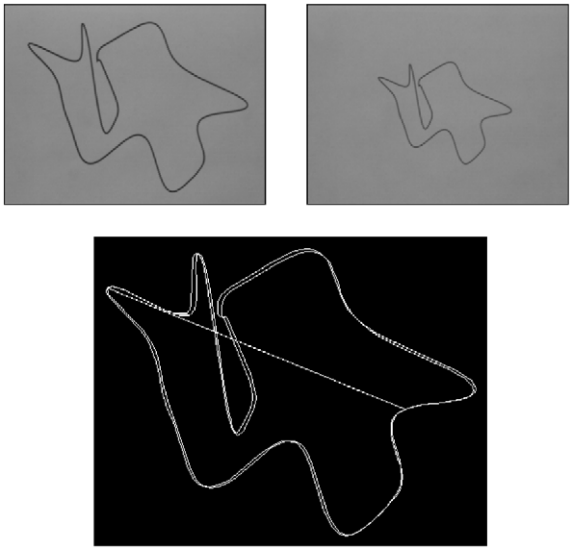

Fig. 20. Matching a more complicated curve: (a) model, (b) object, and (c) projection of (a) to (b).

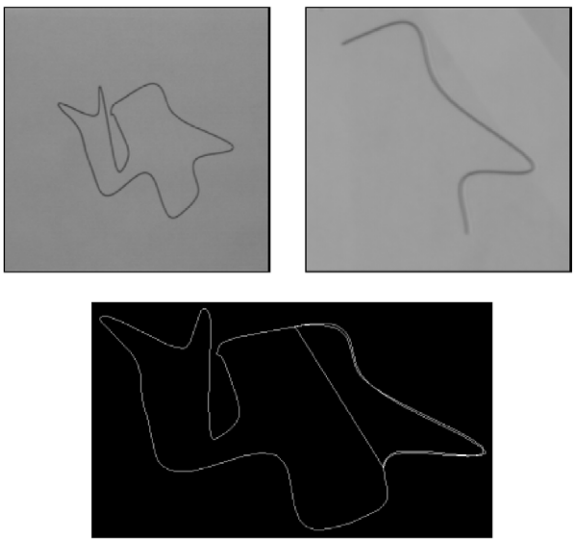

Fig. 21. Matching a segment of the curve to the full contour: (a) model, (b) object, and (c) projection of (a) to (b).

consistent with the experimental settings. The best matching position is shown in Fig. 21, in this case using the percentage of well-matched segments as described in Section 4.6.

6. Conclusions

We have presented a method for matching planar contours viewed from an arbitrary direction in 3D space using the weak perspective projection model. The approach is scale and orientation independent, and can be applied to match both partial and full curves, applying different criteria to a mapping error that is a measure of how well the two contours match. The elements of the affine projection matrix are recovered, and, based on this matrix, it is possible to establish a point-to-point mapping from one curve to the other.

Features points on the curve may be defined as curvature or inflection points on a smoothed $\theta(s)$ coded contour, and these can be used to accelerate the matching process. However, the approach has also been applied successfully to "featureless" contour data. The processing of model and object images is identical, proceeding from an edge image to derive the $\theta(s)$ code that is used as the basis for curve matching. Hence, the approach can be used equally well for matching different camera views, for example as part of a multi-camera or video sequence analysis.

7. Summary

Planar curves provide a rich source of information about a scene. For example, many objects have planar or near-planar features on their surfaces, and the outline of a 3D object can often be treated as a planar curve. These curves are usually, but not necessarily, formed from linked edges extracted from an intensity image. "Virtual" planar curves can also be formed by linking feature points, for example to form the skeleton or medial axis transform of an object.

In this paper, we present a new approach to match planar curves using the weak perspective projection model. This is based on a set of shape parameters that can be extracted from a closed or open contour, derived from the original image as a $\theta(s)$ boundary code. In order to reduce the complexity and increase the robustness of the matching process, the original

parameters are reduced to a set of three intermediate variables, each of which can be calculated independently. These variables are contained within a system of linear equations which define the angles and the ratio of the heights of corresponding point pairs on the two contours with respect to a floating coordinate system. The shape matching process is scale and orientation independent, and the original parameters that describe the relative pose of the two contours in 3D space can be recovered subsequently. The approach can be applied to "featured" and "featureless" contours, to whole and partial contours, and is demonstrated on images of contours and mechanical parts and tools to recover identity and pose.

References

[1] H. Freeman, On the encoding of arbitrary geometric configurations, IRE Trans. Electron. Comput. 10 (2) (1961) 260–268.
[2] H. Yuen, A DM-like differential chain code for handwriting and line graphics, IEE Proc. Vision Image Signal Process. 143 (2) (1996) 96–100.
[3] W.A. Perkins, A model-based vision system for industrial parts, IEEE Trans. Comput. C-27 (2) (1978) 126–143.
[4] Y. Lin, J. Dou, H. Wang, Contour shape description based on an arc-height function, Pattern Recognition 25 (1) (1992) 17–23.
[5] D.H. Ballard, Generalizing the Hough transform to detect arbitrary shapes, Pattern Recognition 13 (2) (1981) 111–122.
[6] C.F. Olson, Improving the generalized Hough transform through imperfect grouping, Image Vision Comput. 16 (9–10) (1998) 627–634.
[7] C.C. Lin, R. Chellappa, Classification of partial 2-d shapes using Fourier descriptors, IEEE Trans. Inf. Theory 9 (5) (1987) 686–690.
[8] S.J. Dickinson, A.P. Pentland, A. Rosenfeld, From volumes to views: an approach to 3D object recognition, CVGIP: Image Understanding 55 (2) (1992) 130–154.
[9] K. Arbter, W.E. Snyder, H. Burkhardt, Application of affine-invariant Fourier descriptors to recognition of 3D objects, IEEE Trans. Pattern Anal. Mach. Intell. 12 (3) (1990) 640–647.
[10] I. Weiss, Review: Geometric invariants and object recognition, Int. J. Comput. Vision 10 (3) (1993) 207–231.
[11] E. Rivlin, I. Weiss, Local invariants for recognition, IEEE Trans. Pattern Anal. Mach. Intell. 17 (3) (1995) 226–238.
[12] A. Branca, E. Stella, A. Distante, Feature matching constrained by cross ratio invariance, Pattern Recognition 33 (3) (2000) 465–481.
[13] D. Cyganski, R.F. Vaz, A linear signal decomposition approach to affine invariant contour identification, Pattern Recognition 22 (12) (1995) 1845–1855.
[14] K.S. Roh, I.S. Kweon, 2D object recognition using invariant contour descriptors and projective refinement, Pattern Recognition 31 (4) (1998) 441–455.
[15] F. Mokhtarian, S. Abbasi, Shape similarity retrieval under affine transforms, Pattern Recognition 35 (1) (2002) 31–41.
[16] F. Mokhtarian, A.K. Mackworth, A theory of multiscale, curvature-based shape representation for planar curves, IEEE Trans. Pattern Anal. Mach. Intell. 14 (8) (1992) 789–805.
[17] M. Kass, A. Witkin, D. Terzopoulos, Snakes: active contour models, Int. J. Comput. Vision 1 (4) (1988) 321–331.
[18] N.S. Friedland, Z. Rosenfeld, An integrated approach to 2D object recognition, Pattern Recognition 30 (3) (1997) 525–535.
[19] J. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. 8 (6) (1986) 679–689.
[20] L.J. Latecki, R. Lakamper, Application of planar shape comparison to object retrieval in image databases, Pattern Recognition 35 (1) (2002) 15–29.
[21] C. Urdiales, A. Bandera, F. Sandova, Non-parametric planar shape representation based on adaptive curvature functions, Pattern Recognition 35 (1) (2002) 43–53.
[22] B. Liang, Matching planar contours and polarimetric analysis of image data, Ph.D. Thesis, Heriot-Watt University, 2001.
[23] Y. Aloimonos, Perspective approximations, Image Vision Comput. 8 (3) (1990) 179–192.
[24] R. Basri, Paraperspective = affine, Int. J. Comput. Vision 19 (2) (1996) 169–179.

About the Author—BOJIAN LIANG received his B.Sc. degree in Communication Engineering from Northern Jiatong University in China in 1982 and his Ph.D. degree in Computer Science from Heriot-Watt University in 2002. Recent interests include object recognition and the application of polarization analysis to computer vision problems. Currently, he is working with the Advanced Computer Architectures Group on robot navigation and neural architectures at the University of York.

About the Author—ANDREW WALLACE was born in St. Andrews, Scotland, in 1951. He graduated in 1972 with a B.Sc. in Electrical and Electronic Engineering, and with a Ph.D. in 1976 for his work on semiconducting chalcogenide glasses, both from Edinburgh University. Since then he has worked on many aspects of computer vision, particularly object recognition, laser imaging and processing, and parallel algorithms. He is currently supported by a Royal Society Industrial Fellowship to further his work in these areas. He is a member of the IEE.