

Robust Recovery of 3D Ellipse Data

Stephen Pollard and John Porrill

AI Vision Research Unit, University of Sheffield
Sheffield, England.

Abstract

This paper is concerned with robust, accurate and computationally tractable methods for the automatic recovery of 3D ellipse data from edge based stereo. The processing paradigm relies heavily on the 2D image as a rich and robust source of scene feature hypotheses (in this case ellipses). Rather than attempt to recover 3D scene descriptions by grouping unstructured estimates of disparity and/or depth, a processes of automatic 2D feature hypothesis is used in conjunction with an appropriate disparity grouping constraint (in the case of ellipse hypotheses we use an affine disparity plane constraint) to recover more accurate 3D scene descriptors.

1 Introduction

Exploratory work in [1] and [2] has shown the recovery of accurate ellipse data in 2D from pure edge data and in 3D from edge based stereo respectively. Here we concentrate attention upon the difficult problem of resolving the scene segmentation problem. That is the task of decomposing the continuous noisy edge contours recovered from the scene in terms of their constituent elliptical conic sections and the subsequent combination of ellipse data recovered from different contours into more complete and useful descriptions.

2 Conic fitting

We are given a string of edge points in 2D with some given accuracy of measurement and we wish to find the best estimate of the conic fitting these points. We represent a conic by its quadratic form

$$ax^2 + 2bxy + cy^2 + 2dx + 2ey + f = 0$$

the 6 coefficients are determined only up to a constant factor. Since we are not interested in fitting hyperbolae we normalise the coefficients using the additional linear constraint

$$a + c = 1$$

so that the conic is completely described by the measurement equation

$$z = \mathbf{h} \cdot \mathbf{x}$$

where

$$\mathbf{x} = (a \quad b \quad d \quad e \quad f)^t$$

$$\mathbf{h} = (x^2 - y^2 \quad 2xy \quad 2x \quad 2y \quad 1)^t$$

$$z = -y^2$$

Assuming that a given measurement point lies on the conic gives us a single linear equation for the conic coefficients. Given five points on the conic we can determine it uniquely (given the above constraint). Given a string of error prone measurements we can determine the least squares solution by standard methods. Though computationally cheap, the result is often highly unsatisfactory, showing a marked bias to high curvature conics. This problem is resolved by the use of the bias corrected linearised Kalman filter [1]. (Using the usual formulation of the linearised Kalman filter does not lead to an improvement, for most implicit equations the linearisation involved takes place about the *wrong point*).

Another advantage of the Kalman filter approach is that it supplies estimates of the variance and covariances of the estimated conic coefficients, given the known accuracy of the data, in the form of a 5 by 5 covariance matrix S . This allows us to test the likelihood that the conic fitted to a small section of data extends to pass through a given point (x, y) by treating the weighted square error

$$\frac{(z - \mathbf{h} \cdot \mathbf{x})^2}{\mathbf{h}' S \mathbf{h}}$$

as a chi squared variable on one degree of freedom. Using this test we can find all areas into which a search for extensions to a fitted conic is reasonable at some confidence level.

3 Edge contour segmentation

The task of decomposing extended edge contours into meaningful subsections has received considerable attention from both the vision and graphics communities. In essence two methods exist. First those that perform an analysis of the local curvature along an edge with a view to identifying points of rapidly changing curvature to act as segmentation or knot points[3][4]. The sections between knot points are then fitted with an appropriate approximating curve. As the curvature computation relies upon the ratio of discrete first and second difference operators it has the effect of amplifying noise. Hence the segmentation is inherently unstable and are unable to overcome errors that arise in the imaging process and are added as a result of edge detection .

We prefer to adopt the alternative strategy in which we fit primitives directly to those parts of the edge contours that lie between knot points without first making the knot points explicit. As we are fitting directly to the edge data it is straightforward to ensure that the method is robust to local deviations of the underlying edge contour from the fitted primitive. Our method has been used previously to obtain robust polygonal approximations [2], it has now been extended to allow generic decomposition of edge contours into any algebraic form.

Before describing our algorithm we shall outline 4 important criteria that affect the segmentation process. **Robustness**: the effect of local noise upon the decomposition process should remain local; preferably the method should be immune to a large class of normally expected errors. **Stability**: the partitioning should be stable and invariant.

Compactness: the decomposition should be concise. The minimal number of approximating primitives should be used. **Efficiency:** the method should be computationally efficient. Preferably the computational complexity should reflect the physical complexity of the edge contour rather than some other implementation dependent metric, such as its length in pixels.

In common with [4] and [5] we believe that an important goal in the design of a segmentation algorithm is that it should recover as close as possible a description of a contour at its most 'natural' scale or scales. Only then can the requirements for robustness, stability, compactness and efficiency be addressed satisfactorily. We differ from previous work in (1) that we try to describe contours by primitives at the largest scale that gives a sufficiently accurate description, and (2) in the nature of the algorithm chosen to perform the segmentation.

3.1 Finding ellipses at a single scale

Imagine that we have sampled an edge contour, that includes at least one elliptical section (e.g. look ahead to figure 1), at a scale 'natural' for segmenting out a particular ellipse. If we consider each set of 5 successive sample points along the contour as the 'starting point' of a potential ellipse then a 'natural' sampling frequency is one for which at least one 5 point span along the contour is guaranteed to almost cover the entire elliptical section.

The algorithm to identify the ellipse is straight-forward. Start by "testing" each 5 point span along the contour to see which if any are appropriate for description as an ellipse. The "test" is performed between the fitted ellipse and the underlying edge contour. The 5 point span sections of the contour that pass the initial ellipse decomposition phase then undergo an iterative refinement process during which the fitted ellipse is 'grown' in each direction along the contour. Each iteration of the growing procedure has three stages:

- i. Apply the bias corrected Kalman filter to all the edge data covered by the current ellipse fit. This results in a more accurate ellipse description.
- ii. Test that the fitted conic is still consistent with the underlying edge data. If not go back to the previous estimate.
- iii. Extend the fitted conic in each direction along the edge contour up to the point where it no longer fits.

The ellipse recovery process completes when no further extension is possible.

We employ a form of hysteresis thresholding in both the testing and growing phases of our segmentation scheme. Three parameters control this process, an upper threshold that must not be exceeded, a lower threshold that can be exceeded transiently and a deviation

count threshold that determines the maximum number of steps for which the lower threshold can be exceeded. Through the use of a large upper threshold and deviation count the segmentation scheme can be made relatively insensitive to moderate local deviations from the true ellipse and at the same time employing a small lower threshold ensures that a tight rein is kept upon slow but systematic deviation from the fitted data.

3.2 Segmenting edge contours at all scales

Edge contours are recursively decomposed into geometric primitives starting from the largest and leading to the smallest using a sophisticated form of split and merge algorithm (although merging is replaced by growing). The algorithm is first applied at a coarse scale of description and then recursively re-applied at ever finer sub-scales until all sections of the edge contour have either been described or the remaining fragments fall below a threshold for that type of primitive.

The recursive procedure to describe an edge contour has the following form:

```

contour_segment(start, end, sample, step)
ContourPoint      start
ContourPoint      end
IntegerNumber     sample
IntegerNumber     step
{
  [1] consider in turn each span, of length "sample", along
      the contour until a suitable fit is found between the
      chosen primitive and the underlying edge contour (this
      could be the best such fit).

  [2] if the fitting process was successful then grow the
      primitive along the edge contour keeping track of its
      start and end (pstart and pend respectively).
      Now recursively call
      contour_segment(start, pstart, sample, step)
      contour_segment(pend, end, sample, step)
      to describe the remaining sections of the contour and
      EXIT this function call.

  [3] reduce the sample and step lengths by a fraction.

  [4] if the sample is now below a sample length threshold
      then no fit is possible for this contour section so EXIT
      this function call.

  [5] recursively call
      contour_segment(start, end, sample, step)
      with the reduced sample length and step size.
}

```

In this recursive procedure the arguments **start** and **end** define the two end points of the edge contour or a contiguous sub-contour extracted from it. The argument **sample** is the span length parameter that defines the current sample frequency and **step** determines the degree of overlap of neighbouring samples. The effect of a call to this procedure is to

keep increasing the initial sampling frequency by reducing the sample length until it is possible to get a satisfactory fit between the primitive and a contour sub-section. This is the largest one (or the first equal largest one) and is described at approximately its natural scale. In practice it is admissible to impose an upper threshold on the sample length without unduly affecting the behaviour of the algorithm. The recovered primitive is then extended along the contour within the bounds defined by **start** and **end**. The procedure then deals in turn with the remaining sub-sections of the contour.

Despite the apparent algorithmic complexity of this approach the method is in fact computationally effective. The complexity of the algorithm is independent of the length of the edge contour *per se*; for example two single span ellipse sections of different lengths will be recovered in the same time. It depends instead upon the physical complexity of the contour. As geometrical primitives are identified at approximately their largest natural scale repetitive work at finer scales is avoided. Hence the algorithm has a favourable computational workload when compared to other algorithms used for contour decomposition. Furthermore, fitting and growing large sections first also improves the robustness of segmentation. Short but convoluted contour sections do not effect this algorithm.

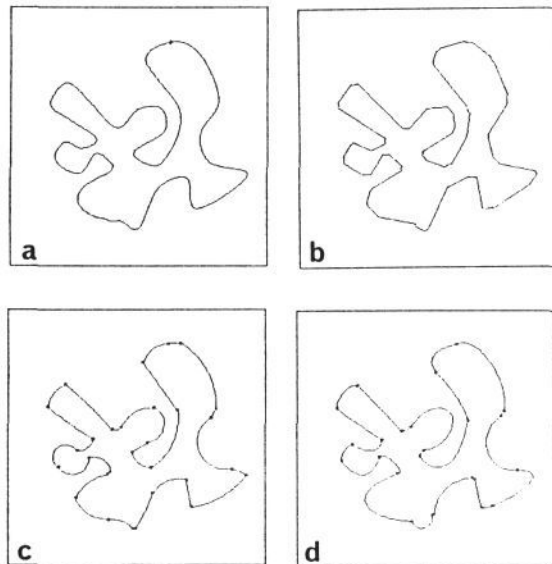


Figure 1. Example of segmenting a complex edge string. The edge contour shown in (a) was recovered from a real image using a local implementation of the Canny edge detector and a nearest neighbour edge linking algorithm. The decompositions in (b), (c) and (d) are obtained with straight line, circle and ellipse based segmentation schemes respectively. The knot points are used to illustrate the less obvious segmentations of (c) and (d). All 3 segmentations used the same standard set of parameter and thresholding values.

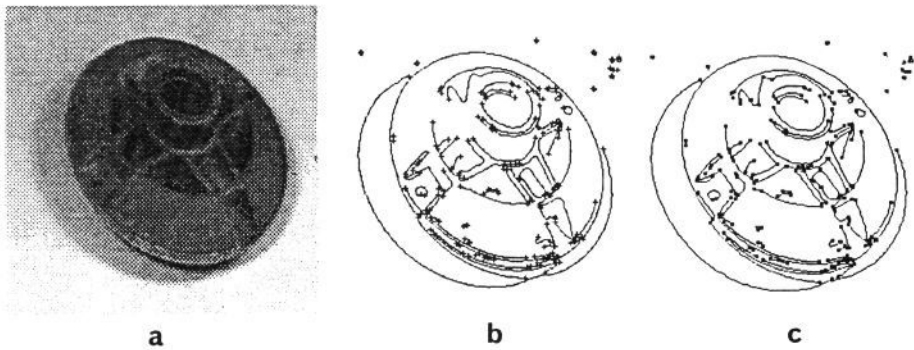


Figure 2. *Example of an industrial part. The theme object used throughout the remainder of the paper is shown on (a), with Canny edge contours shown in (b), and segmented ellipses in (c).*

4 Feature combination

We now address the issue of feature combination. That is, which elliptical curves can be combined with other ellipses or straight lines to produce larger and more robust descriptions. Larger primitives form a better basis for subsequent integration with disparity data and result in more accurate 3D primitives. However, improper grouping at this stage may result in the creation of accidental 2D features that have no 3D counterparts. Hence we adopt a fairly conservative grouping strategy.

The principal criteria for allowable grouping is the statistical test provided by the confidence interval of the Kalman filter. As a hard and fast rule we reject for combination those primitives whose mid-points lie outside the 99% confidence limit with respect to the ellipse that is under consideration for extension. Based upon this very liberal initial criteria we have developed a rapid index into the set of possible primitives that could be used to extend the current ellipse. First we produce a spatial index of the visual array on the basis of a regular grid of bins into each of which we store the list of geometrical primitives whose mid-points lie within it. Hence each feature is represented exactly once in the indexing array. Clearly it is now possible to use the chi-square test with respect to the grid points of the index to determine which lists of image features are candidates for inclusion in the extension. If either the extrapolated ellipse (and not the physical part of the ellipse already recovered) passes through a bin of the index grid or one or more vertices of that bin lies within the 99% confidence limit then the list of primitives whose mid-points are stored within it are chosen as potential candidates. What we have in effect is a quantised form of the confidence envelope of the original ellipse.

Rather than exhaustively searching the whole spatial index for just those few grid elements that fall within the confidence envelope, we exploit the monotonic nature with which confidence falls off with respect to distance away from the extrapolated ellipse. In other words the confidence envelope is fully connected and once we have identified a single grid element that lies within the confidence envelope we know it must be indirectly connected in the eight neighbour sense to all other such elements. Points along the extrapolated curve (especially its midpoint) provide prime and easily computable candidates for a simple bush-fire technique to identify all elements inside the envelope (see figure 3).

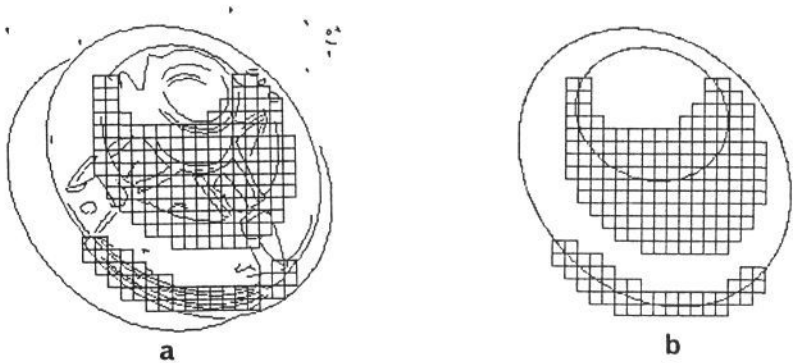


Figure 3. *Confidence envelope indexing. The results of using the quantised confidence envelope indexing scheme are illustrated for 2 partial ellipses identified in the theme figure. In (a) the indexed members of the spatial array are shown superimposed over the original ellipse data. In (b) the ellipses that gave rise to them are shown in isolation, extended along the loci of highest confidence recovered by the Kalman filter. Notice how the longer more complete ellipse provides a much tighter constraint envelope.*

In addition to high values of chi-square confidence (90% for example) we have identified a number of additional heuristics that help select good candidates for possible conic extension. They are:

Proximity. That the closest end-points of the current ellipse and the candidate feature fall within a small threshold.

Co-tangency. That the tangent directions of the closest end-points of the current ellipse and the candidate feature are almost collinear.

Exclusivity. Curve segments must not overlap along their lengths.

Maximum Gap. Do not grow a conic over a gap that is larger than either one of the constituent parts.

Invariant Curvature Ratio. Any two points on the same ellipse should have an invariant curvature ratio of almost unity. This ratio is defined as:

$$\frac{\kappa_1 \sin^3 \theta_2}{\kappa_2 \sin^3 \theta_1}$$

where κ is curvature and θ is the angle between the chord (between the 2 points) and the tangent.

Consistency. Is the combined ellipse consistent with each of the constituent edge contours.

Note that in order to deal with slight over extension of the original ellipses that may have occurred during their growing phase (especially if they are not bounded by tangent discontinuities) we only insist upon consistency with respect to the combined conic description away from the endpoints of the original edge contours (see figure 4).

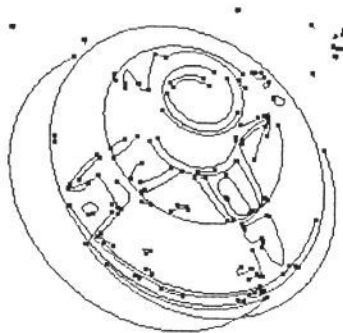


Figure 4. *Combined conics.*

5 Plane fitting

As we perform stereo matching and conic fitting in a 'rectified' parallel camera coordinate frame that is equivalent to the original camera geometry (which is itself recovered by accurate calibration procedure) then the transform of a plane curve between the images will be a y preserving affine as all epipolars are horizontal. That is, the transform from left to right image points is given by:

$$X_r = aX_l + bY + c \quad (1)$$

where X_l and X_r are the left and right horizontal components of position and Y is the common vertical component.

Rearranging we get

$$X_r = \frac{X_l}{a} - \frac{bY}{a} - \frac{c}{a} \quad (2)$$

Using (1) we can solve for the affine parameters using the weighted least squares form of:

$$\begin{pmatrix} X_r & Y & 1 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} X_l \\ \cdot \\ \cdot \\ \cdot \end{pmatrix}$$

Disparity D is defined as $X_r - X_l$, hence from (2):

$$D = X_l \left(\frac{1}{a} - 1 \right) + Y \left(-\frac{b}{a} \right) - \frac{c}{a}$$

Which defines a plane in a disparity space based upon the left image.

$$D = X_l A + Y B + C$$

From which it is straight forward to recover the plane in world coordinates based upon the optical centre of the virtual 'rectified' left camera.

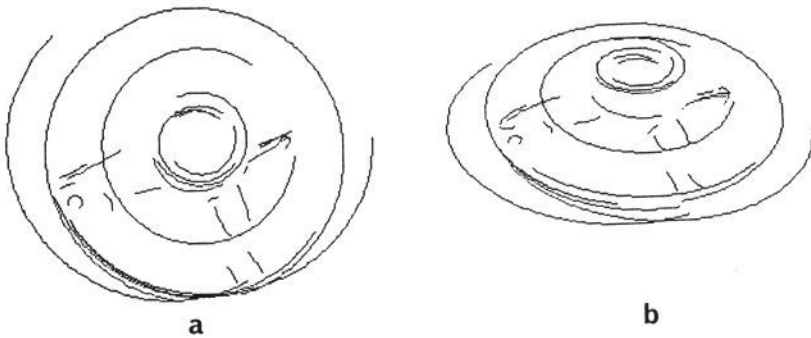


Figure 5. Two very different views of the reconstructed 3D ellipse data recovered from our theme image are shown in (a) and (b).

6 Conclusions

We have shown that it is possible to exploit the rich source of structural information that exists in the 2D image in order to drive the recovery of 3D information. The resulting 2D and 3D geometrical primitives are both robust and accurate. Furthermore the algorithms that we have exploited are computationally feasible on modern day computer hardware. Our emerging competence at representing and from there recognising complex industrial parts goes some way to reducing the gap between that exist between state of the art computer vision systems and the requirements of complex visual automata for the industrial market-place. However, extensions to the work presented here could and probably should be directed along the following 3 directions.

1. Introduction of higher level grouping constraints in both 2D and 3D. Examples include co-centricity, co-linearity, co-planarity, orthogonality, cylindricality, etc.
2. Exploitation of more data from the 2D image. Currently we restrict our image based grouping and description operators to work upon edge data alone.
3. Two eyes are better than one. Ideally the stereo description should tend towards the union rather than the intersection of the information provided by each image.

To summarise, the act of forcing a unique description for a particular section of an edge string is not always a reliable strategy. Only in the context of higher level descriptions (e.g. full conics, cylinders, etc.) which are verified against the underlying image intensity data can some lower level ambiguities be resolved. The descriptions obtained in this paper can be used to provide a powerful initial hypothesis for such a strategy.

References

- [1] Porrill J (1990) Fitting ellipses and predicting confidence envelopes using a bias corrected Kalman filter, *Image and Vision Computing*, vol. 8, no 1, 37-41.
- [2] Pollard S B, Porrill J, and Mayhew J E W (1991) Recovering partial 3D wire frames descriptions from stereo data, *Image and Vision Computing*, vol 9, no 1, 58-65.
- [3] Asada H, and Brady M (1986) The curvature primal sketch, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, no 1, 2-14.
- [4] Wuecher D M and Boyer K L (1991) Robust contour decomposition using a constant curvature criterion, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-13, no 1, 41-51.
- [5] Hoffman D D (1983) Representing shapes for visual recognition, PhD Thesis Massachusetts Institute of Technology.
- [6] Lowe D G (1988) Organisation of smooth image curves at multiple scales, in *Proc. 2nd Int. Con. Computer Vision*, 558-567.