Polygonal Approximation of Closed Curves across Multiple Views

M. Pawan Kumar Saurabh Goyal C. V. Jawahar P. J. Narayanan
Centre for Visual Information Technology
International Institute of Information Technology
Gachibowli, Hyderabad 500 019
{pawan@gdit., saurabh@gdit. jawahar@, pjn@}iiit.net

Abstract

Polygon approximation is an important step in the recognition of planar shapes. Traditional polygonal approximation algorithms handle only images that are related by a similarity transformation. The transformation of a planar shape as the viewpoint changes with a perspective camera is a general projective one. In this paper, we present a novel method for polygonal approximation of closed curves that is invariant to projective transformation. The polygons generated by our algorithm from two images, related by a projective homography, are isomorphic. We also describe an application of this in the form of numeral recognition. We demonstrate the importance of this algorithm for real-life applications like number plate recognition, aircraft recognition and metric rectification.

1. Introduction

We recognize a large number of familiar and novel objects every day with little effort. We can also recognize many objects that may vary significantly in form when viewed from different view points. Objects can be recognized even when they are partially obstructed. The recognition of objects from different views is a well studied problem in computer vision. Most of the literature, however, is confined to similarity transformations between different views, and not the general case.

Planar objects are subset of objects but of great practical interest. Polygonal approximation – i.e, approximating a given closed curve as a 2D polygon – provides a simple representation of the planar object boundary. Parameterizing the boundary using a polyline representation makes recognition easy. Polygonal approximation has also been used as an intermediate step in various applications such as volume rendering and multiresolution modeling [3, 12].

Let $X = \{x_1, x_2, \ldots, x_n\}$ be a set of points on the boundary of a planar object to be approximated using a polygon. Polygonal approximation can be defined as a partitioning of the set into k mutually exclusive and collectively exhaustive subsets $\phi_1, \ldots, \phi_k$ such that each of the subsets can be approximated using a linear sequence of points. This can be achieved by minimization of an objective function of the form:

$$J = \sum_{i=1}^{n} d(x_i, l_j), \quad x_i \in \phi_j \tag{1}$$

where l_j is the linear structure which approximates the points in ϕ_j and $d(\cdot)$ is a measure of deviation of x_i from l_j. The deviation $d(\cdot)$ can be the perpendicular distance or another measure of how the linear structure l_j distorts the real point x_i. Thus, the deviation can be considered as the error in fitting the linear structure to the set of points. Given ϕ, problem of identification of l can be addressed using classical regression techniques or eigen vectors of the covariance matrix [4]. A general optimization of the objective function may be computationally expensive and prone to get stuck in local minima. Therefore, most of popular polygonal approximation algorithms look for an optimal solution with the help of a greedy algorithm [6, 9]. They primarily exploit the advantage that the points in the set are connected and ordered. Dynamic Strip Algorithm is one such algorithm [6] which is a fast polygonal approximation algorithm. Hopfield neural network based algorithms also have been reported for polygonal approximation [2]. Some methods do not exploit the connectedness of points [1]. They group points on the boundary into linear clusters. Some of the algorithms emphasize the optimality and efficiency of the polygon approximation [11].

The above mentioned algorithms are primarily designed to handle similarity image transformations – mostly, transformations involving translation, rotation, and scaling – for recognition. They do not perform well if the two boundaries being compared differ more than by a similarity transformation. When a planar object is imaged from multiple viewing positions the image-to-image transformation is projective, a lot richer than the similarity transformation [5].

In this paper, a polygonal approximation algorithm for closed curves which is invariant to projective transformation is proposed. We explore its application as an intermediate step in planar object recognition where it particularly useful as it avoids the problem of correspondence on the object contour. In Section 2, we formally introduce the problem. In Section 3, a description of a polygonal approximation algorithm which is invariant to projective transformation is presented. We discuss the implementation details of the algorithm in Section 4. Section 5 presents the results of the algorithm and describes real-life applications of the proposed algorithm.

2. Preliminaries

2.1. Image-to-Image Homographies

When an object is imaged from multiple viewpoints, points on it undergo a transformation. The transformation that the coordinates of each point of a plane undergo from one image to the other can be mathematically described as a general projective or linear operation in homogeneous coordinates. In some special cases, the relation could have simpler forms of affine or similarity transformation.

It has been shown that a planar object viewed from multiple viewing positions results in a projective image-to-image homography.

2.2. Problem Formulation

Let $\{x_1, x_2, \ldots, x_n\}$ be the points on the boundary of an object in the first view. To approximate these points using a polygon, one can follow the same procedure discussed in the previous section by minimizing an objective function same or similar to Equation 1.

Let y_i be the point corresponding to x_i in another view. Then, $y_i = \mathbf{T}x_i$. The polygonal approximation algorithm working on the second view minimizes the objective function

$$J' = \sum_{i=1}^{n} d(y_i, l'_j), \quad y_i \in \phi'_j \tag{2}$$

where l'_j is the linear structure which approximates the points in ϕ'_j. For the polygonal approximation to be invariant to projective transformation, there should be a one-to-one mapping from ϕ'_i to ϕ_i. We address this problem using projective invariants.

2.3. Projective Invariants

An invariant, $I(p)$, of a geometric structure described by a parameter vector p subject to a linear transformation T of the coordinates $x' = Tx$, is transformed according to $I(p') = I(p)|T|^w$ [8]. Here $I(p)$ is the function of the parameters after the linear transformation. Invariants for which $w = 0$ are referred to as scalar invariants [8]. Consider four points p_1, p_2, p_3, p_4 and their projective transformations as shown in Figure 1. Neither distances nor ratio of distances are preserved in this case. Below we describe two projective scalar invariants.

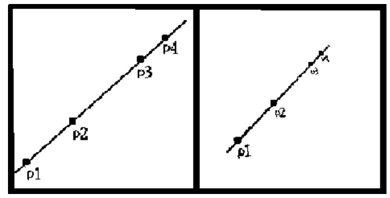

Figure 1: Four collinear points before and after undergoing projective transformation

Cross-ratio of four collinear points: The cross-ratio of points p_1, p_2, p_3, p_4 is defined as

$$cr(p_1, p_2, p_3, p_4) = \frac{(X_3 - X_1) \cdot (X_4 - X_2)}{(X_3 - X_2) \cdot (X_4 - X_1)}, \tag{3}$$

where X_1, X_2, X_3, X_4 represent the corresponding positions of each point along the line. $(X_3 - X_1)$ is the distance between points p_3 and p_1. This is invariant to a general projective transformation [8].

Cross-ratio of areas of five points: The cross-ratio of the areas is defined by

$$cr(p_1, p_2, p_3, p_4, p_5) = \frac{\triangle_{p'_1 p'_2 p'_5} \cdot \triangle_{p'_3 p'_4 p'_5}}{\triangle_{p'_1 p'_3 p'_5} \cdot \triangle_{p'_2 p'_4 p'_5}} \ , \tag{4}$$

where $\triangle_{p_1 p_2 p_5}$ is the area of the triangle formed by points p_1, p_2, p_5. This is invariant to general linear or projective transformations [8].

3. Projective Invariant Polygon Approximation

We first define a ratio of cross-ratios of areas, denoted by λ. Consider points x_1, x_2, x_3, x_4, x_5 and x_6. We have,

$$cr_{12345} = \frac{\triangle_{541} \cdot \triangle_{321}}{\triangle_{531} \cdot \triangle_{421}}, \quad cr_{12346} = \frac{\triangle_{641} \cdot \triangle_{321}}{\triangle_{631} \cdot \triangle_{421}}.$$

We see that the area $\triangle_{541} = \frac{1}{2} \cdot \text{base} \cdot \text{height} = \frac{1}{2} \cdot len_{12} \cdot dist_5^{14}$, where len_{12} is the length of the line segment defined by points 1 and 2 and $dist_1^{23}$ is the perpendicular distance of point 1 from the line defined by points 2 and 3. We define a ratio of cross-ratios as

$$\begin{aligned} \lambda &= \frac{cr_{12345}}{cr_{12346}} = \frac{\triangle_{541} \cdot \triangle_{631}}{\triangle_{531} \cdot \triangle_{641}} \\ &= \frac{len_{14} \cdot dist_5^{14} \cdot len_{13} \cdot dist_6^{13}}{len_{13} \cdot dist_5^{13} \cdot len_{14} \cdot dist_6^{14}} \\ &= \frac{dist_5^{14} \cdot dist_6^{13}}{dist_5^{14} \cdot dist_6^{13}} \end{aligned}$$

which can be rewritten as

$$\lambda = \frac{dist_5^{14}/dist_5^{13}}{dist_6^{14}/dist_6^{13}}. \quad (5)$$

We first establish three useful properties before proving the existence of projective invariant polygon approximation.

Lemma 1: An invariant polygonal approximation algorithm can exist only for a transformation where collinearity is preserved.
Proof: We prove this by contradiction. Say we have a polygonal approximation algorithm invariant to a transformation which does not preserve collinearity. Under such a transformation, not every line gets transformed as a line. From this it is clear that there cannot exist a polygonal approximation algorithm which is invariant to such a transformation. □

Since projective transformation preserves collinearity, there can exist a polygonal approximation algorithm invariant to this transformation. We now establish that the measure, ratio of cross-ratios of area, can be used as the measure of deviation $d(\cdot)$ in Equation 1.

Lemma 2: The ratio of cross-ratios of areas can give us a measure of deviation $d(\cdot)$ in the objective function of polygonal approximation given by Equation 1.
Proof: It is clear from Equation 5 that λ can be interpreted as the ratio of the ratio of perpendicular distances of the points p_5 and p_6 from the two lines $line_{13}$ and $line_{14}$. If this quantity is equal to 1, the two ratios of perpendicular distances are equal. The locus of all such points is the straight line $line_{15}$. When the point p_6 moves away from this line, λ moves away from 1. Thus, λ gives a measure of the collinearity of the points p_1, p_5 and p_6. A simple measure $d() = |\lambda - 1|$ can then be used as the measure of deviation as given in Equation 1. □

Another constraint for the existence of a polygonal approximation algorithm invariant to projective transformation is that the distance function $d(\cdot)$ should be invariant to projective transformation.

Lemma 3: The ratio of cross-ratios of area (λ) is invariant to projective transformation.
Proof: The result follows since the cross-ratios of areas are projectively invariant. □.

We now construct an algorithm which will minimize the optimization function with $d()$ as defined above. This algorithm may be computationally expensive. We can additionally exploit the connectedness and the order of the points of the boundary. This results in an $O(n)$ algorithm for polygonal approximation where n is the number of points on the boundary. We traverse the boundary in a clockwise direction and successively insert points into the set ϕ_i until we encounter a point which deviates too much from the current linear structure l_j, measured using $|\lambda - 1|$.

Theorem 1: There exists an algorithm for polygonal approximation which is invariant to projective transformation.
Proof: From the above lemmas, it is clear there is a function $d()$ which is invariant to projective transformation. Also, if $d()$ is invariant, the above algorithm would result in a one-to-one mapping between the subsets $\phi_1, \phi_2, \ldots, \phi_k$. Therefore, the above algorithm is invariant to projective transformation. □

4. Algorithm, Implementation and Discussion

We develop the algorithm in the following manner.

1. Choose point 1 as the starting point and point 5 as the point next to point 1 on the curve. Points 2, 3, and 4 may or may not lie on the curve.

2. We look at the point adjacent to 5 in the clockwise direction on the curve. Call this point 6. We measure the quantity $d(6) = |\lambda(6) - 1|$.

3. If $d(6) < t$ where t is a suitable tolerance threshold, the line joining points 1 and 5 can approximate the curve up to this point. We move the point 6 forward and repeat the process from step 2.

4. Otherwise, the point 6 has deviated from the $line_{15}$ sufficiently. We, therefore, approximate the curve from 1 to the point before 6 by the line joining points 1 and 5. We then repeat the procedure by taking point 6 to be the new point 1.

The algorithm described above was implemented to work with real images. Real images, however, pose many challenges. Points 3, 4, and 5 have to chosen carefully in order to overcome errors that are present in real images. The most prominent error is due to the discretization or pixelization of the curve because of which the cross-ratios calculated may differ in different views. The error associated with computing the cross-ratios from real images is discussed in [8]. In order to decrease the relative error, points 3 and 4 were taken significantly far from point 1 and from each other. Point 5 was chosen to be a fixed number of points after point 1 because for the points near point 1, cross-ratios tend to vary considerably due to discretization. Point 5 need not be the same in multiple views of the same object as long as it is close to point 1.

The reference points should not be collinear as the area based cross-ratios may not be defined for them. Such configurations have to be avoided carefully. Due to discretization, points collinear in one view may not be collinear in another. Therefore, a different algorithm for checking collinearity was used which does not require points to be strictly collinear. Another problem arises due to the differences in the location of point 3 and 4 in different views. However, as long as the threshold is low, this results in a difference of only 1-2 pixels in the boundary points. We have tested real images for various appropriate positions of points 3 and 4 and have found that the λ does not change significantly with changes in points 3 and 4 and therefore, it acts as a projective invariant which is a good measure of collinearity. The choice of the threshold should depend on the curvature of the section that we are trying to approximate. For example, for a linear section the threshold should be low to retrieve the same linear section as the approximation.

5. Results

The results of the proposed algorithm on two planar boundaries are shown in the Figure 2. The results were obtained on images of size 300×300. We considered only the boundary pixels for our algorithm. The original image is shown on top with two projectively transformed versions below it. The boundary pixels are drawn in black colour. The red lines in the figure show the polygonal approximation of the boundary using our algorithm. The polygon approximation has twenty sides. The number of sides of the polygon and their relative position with respect to the object remain same across all projective transformations we have considered. To quantitatively compare the results, we projectively transformed the polygonal approximation of the original image by the same projective transformations shown as green lines. The green and red polygons in the other figures are identical in all cases except one or two nodes which got shifted by 1-2 pixels due to discretization. Another example is shown on the right in Figure 2. Here an aircraft image is polygonal approximated with a twenty five sided polygon. The boundary of the aircraft is shown in black and the red lines show the polygonal approximation of the boundary. Our experience with other planar boundaries is also very good.

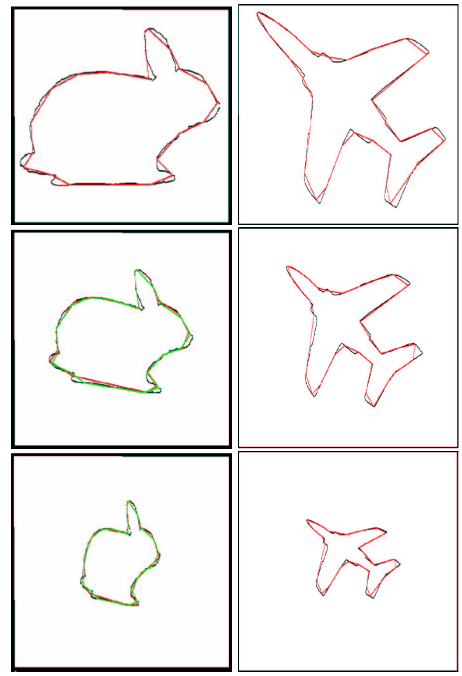

Figure 2: Polygon approximation of two original curves and after applying two projective transformations.

5.1. Numeral Recognition

An important application of polygonal approximation is planar object recognition [7, 10]. We demonstrate the applicability of this algorithm for recognition of numerals across multiple views. Numeral recognition has been conventionally addressed among the document image processing community. There are many other situations in image and video processing where the numerals are to be recognized under projective transformation. Conventional OCRs are not designed to address this problem. We considered numeral images of size 300×300 for this experiment. Thirty four different views of each numeral were considered for experimentation. Thirty of these images used as inputs are shown in Figure 3.

The boundaries of these numerals were extracted and were approximated by a polygon using our algorithm. A feature vector was then formed by taking the cross-ratio of areas of every five consecutive boundary points of the determined polygon. In order to classify the test images, we employed a nearest neighbour classifier. We calculated the Euclidean distance of the feature vector of the test image from the feature vectors of the reference images of each class. The test image was assigned to the nearest class. In a synthetic dataset of size 340 images, we could achieve an accuracy of 94.70%.

Figure 4: Two projectively transformed images of a number plate

This can be extended to recognition of numerals on a number plate. There are many number plate detection algorithms for images and videos. Coupled with this we need a module which can recognize alpha-numerals under projec-

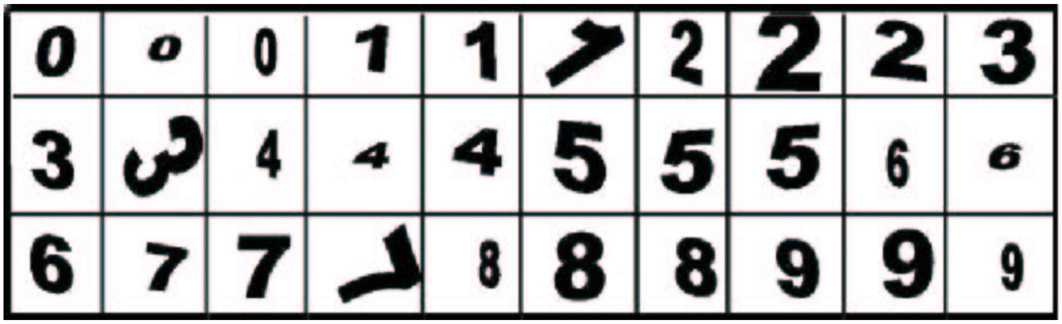

Figure 3: Some of the inputs used in the numeral recognition system

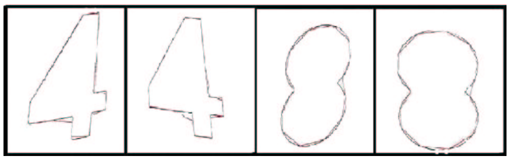

Figure 5: Polygon approximations of real images of 4 and 8 extracted from a number plate

tive transformation. In a number plate recognition system, the input is a projectively transformed image of numerals and alphabets, as shown in Figure 4 .It's objective is to identify the number on the number plate. To test the feasibility of number plate recognition using our algorithm, we analyzed a few real images of number plates. The test was limited to recognition of digits and can be easily extended for alphabets as well. A sample polygonal approximation is shown in Figure 5. We are yet to test this system on a large database of number plates. Table 1 shows the results obtained on various numerals which were obtained from number plates.

Digit	Number of Samples	Accuracy(%)
0	10	100.00
1	10	100.00
2	10	90.00
3	10	80.00
4	10	90.00
5	10	100.00
6	10	100.00
7	10	100.00
8	10	100.00
9	10	90.00
Total	100	95.00

Table 1: Recognition accuracy for the tests on number plates.

5.2. Aircraft Recognition

In an aircraft recognition system, we are given projectively transformed images of aircrafts of various aircraft models for recognition. The system should be able to identify the aircrafts correctly by distinguishing between various models. The polygon approximation of one such aircraft is shown on right side of Figure 2. The system has not been tested with a large database of aircraft images. Table 2 shows the recognition results obtained using images of 15 different types of aircrafts. Some of the sample inputs are shown in Fig 6.

Type of Aircraft	Number of Samples	Accuracy(%)
1	10	90.00
2	10	90.00
3	10	100.00
4	10	100.00
5	10	80.00
6	10	100.00
7	10	100.00
8	10	90.00
9	10	90.00
10	10	90.00
11	10	100.00
12	10	80.00
13	10	80.00
14	10	100.00
15	10	100.00
Total	100	92.66

Table 2: Recognition accuracy for the tests on aircraft images.

5.3. Metric Rectification

Another application of the proposed algorithm is metric rectification. We aim at removing projective and affine distortions from a given image of a plane. In order to achieve this,

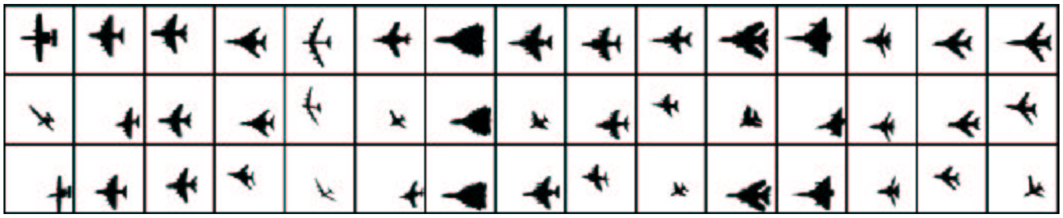

Figure 6: Some of the inputs used in aircraft recognition system

it is enough to identify two pairs of parallel lines and two pairs of perpendicular lines in the image [5].

A circle is imaged as a conic section. The image of one circle doesn't provide enough constraints for metric rectification. Our polygonal approximation algorithm enables us to approximate any conic section with a fixed number of lines. Thus, if we approximate the circle's image using four lines, we obtain two pairs of parallel lines and two pairs of perpendicular lines as circle is approximate as a square using four lines. This will enable us to perform metric rectification. Examples are shown in Figures 7 and 8. Figure 8 shows the synthetic distorted image and the result obtained by metric rectification. Rectification was made by using the circular 'O' in the title of the book.

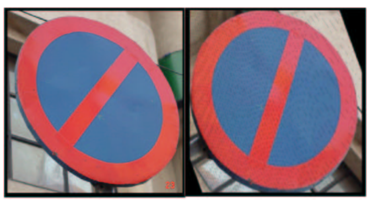

Figure 7: Original Image and Rectified Image of a road-sign

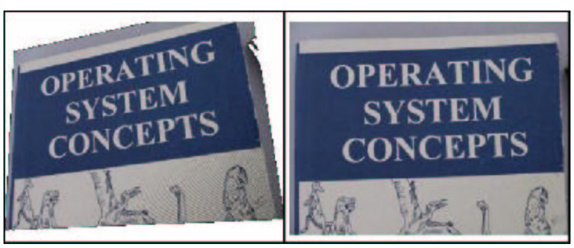

Figure 8: Distorted Image and Rectified Image of a book.'O' in the title was treated as circle.

6. Conclusions and Future Work

We presented an approach to generate a projectively invariant polygonal approximation using invariant properties of the cross-ratio of areas. We demonstrated how planar shape recognition can be achieved using this algorithm. This can be applied to real-life problems such as number plate recognition and aircraft recognition. We plan to use polygon approximation for object recognition with occlusion. Also, other applications have to be explored like shape from texture. Repeated texture elements can be polygon approximated and may be used for reconstruction of shape.

References

[1] I. Anderson and J. Bezdek. Curvature and tangential deflection of discrete arcs. *T-PAMI*, 6:27 – 40, 1984.

[2] P. Chung. Polygon approximation using a competitive hopfield neural network. *Pattern Recognition*, 27(1):1505 – 1512, 1994.

[3] M. DeHaemer, Jr. and M. J. Zyda. Simplification of objects rendered by polygonal approximations. *Computers and Graphics*, 15(2):175 – 184, 1991.

[4] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Willey & Sons, New York, 1973.

[5] R. Hartley and A. Zisserman. *Multiple View Geometry*. Cambridge University Press, 2000.

[6] M. K. Leung and Y. Yang. Dynamic strip algorithm in curve fitting. In *Computer Vision,Graphics and Image Processing*, volume 23, pages 69 – 79, 1990.

[7] M. Lourakis, S. Halkidis, and S. Orphanoudakis. Matching disparate views of planar surfaces using projective invariants. In *British Machine Vision Conference*, volume 1, pages 94 – 104, 1998.

[8] J. L. Mundy and A. Zisserman. *Geometric Invariance in Computer Vision*. MIT Press, 1992.

[9] T. Pavlidis and S. Horowitz. Segmentation of plane curves. In *IEEE Transactions on Computers*, volume C-23, pages 860 – 870, 1984.

[10] C. Rothwell, A. Zisserman, D. Forsyth, and J. Mundy. Planar object recognition using projective shape representation. *IJCV*, 16:57 – 99, 1995.

[11] M. Salotti. An efficient algorithm for the optimal polygonal approximation of digitized curves. *Pattern Recognition Letters*, 22(2):215–221, February 2001.

[12] P. Shirley and A. A. Tuchman. Polygonal approximation to direct scalar volume rendering. In *Proceedings San Diego Workshop on Volume Visualization, Computer Graphics*, volume 24, pages 63 – 70, 1990.