Homography Estimation from Planar Contours

Paresh Kumar Jain, C. V. Jawahar
Centre for Visual Information Technology,
International Institute of Information Technology, Hyderabad. INDIA.
jawahar@iiit.ac.in

Abstract

Homography estimation is an important step in many computer vision algorithms. Most existing algorithms estimate the homography from point or line correspondences which are difficult to reliably obtain in many real-life situations. In this paper we propose a technique based on correspondences of contours. Homography estimation is carried out in Fourier domain. Starting from an affine estimate, the proposed algorithm computes the projective homography in an iterative manner. This technique does not require explicit point to point correspondences; in fact such point correspondences are a by-product of the proposed algorithm. Experimental results and applications validate the use of our technique.

1. Introduction

A homography is a non-singular linear relationship between points in two images [3]. When the world points are on a plane, their images captured by two perspective cameras are related by a 3×3 projective homography $\mathbf{H}$. It is well known that

$$\mathbf{y} = \mathbf{H}\mathbf{x}, \quad (1)$$

where $\mathbf{x}$ and $\mathbf{y}$ are the corresponding points (in homogeneous coordinates) in the first and second view respectively. Points in two images can be related by a unique homography under many other situations (eg. when the cameras for multiple scenes share the optical center).

Numerical computation of homography from image measurements is an important step in tasks like calibration [17], metric rectification, 3D reconstruction, mosaicing [12, 15] etc. In some applications, a simple similarity or affine transformation could suffice but many applications need a more general transformation. Problems like recognition and registration often deal with complete projective transformations. Given four corresponding points or lines in a general configuration, the 3×3 homography matrix ($\mathbf{H}$) can be estimated [3] as each correspondence provides 2 constraints.

Most algorithms in multiple view geometry, which compute algebraic relationships like homographies, need accurate correspondences. Such accurate correspondences are often difficult to obtain in practical situations. Algorithms which use lines or similar higher order primitives have been shown to be better suited for many geometric computations [3] as compared to points. In this paper, we extend the notion of higher order primitive further to include an ordered collection of points – a contour. Compared to corresponding points, corresponding contours can be easily and robustly identified. We present a novel Fourier domain technique to compute the homography between two views that only needs corresponding contours – no explicit point-to-point correspondence is needed. In fact, the point-to-point correspondence is obtained as a by-product of our homography computation scheme. Major claims of this paper are summarized below:

- We argue that an ordered set of pixels (say the contour of an arbitrary planar object) is sufficient for estimating the projective homography between two views.
- Our algorithm is defined on a sequence/collection of points. We employ Fourier domain relationships of sequences for the computation of homography. This avoids explicit pixel-to-pixel correspondences.
- Explicit pixel to pixel correspondences can be obtained as a by-product of our algorithm.
- Our algorithm first computes an affine approximation to the actual homography. This transformation is then used to initialize an iterative procedure that computes the complete projective transformation by estimating the projective depth.
- The proposed algorithm provides accurate projective homographies in few iterations.

Source	Primitive	Technique	Transformation	Remarks
Many	Points, Patches	Correlation, Transform Domain Analysis	Similarity	Popular for image registration. Well studied in image processing literature.
Books and earlier work [3, 14]	Points, Lines	Numerically solving linear equations (DLT)	Projective	Direct closed form solution. Strong dependence on accurate correspondence.
Luong *et al.*, etc. [11, 3]	Points with additional clues	Use of weak calibration	Projective	Use additional clues like Fundamental Matrix, which again needs correspondence for estimation.
Kanatani *et al.*, etc. [4, 2]	Points, Lines etc.	RANSAC, ML, Least Squares Estimates	Projective	Large number of possibly noisy correspondences; More robust than DLT; Very popular.
Kuthirummal *et al.* [10, 6]	Nonparametric contour	Fourier Transform of sequences	Affine	Computes affine invariants and polygonal approximations of contours in Fourier domain.
Kruger *et al.*, etc. [5, 9]	Texture	Fourier Transform of image patches	Affine	Minimal line correspondence; upto affine homographies.
Kumar *et al.*, etc. [8, 6, 13]	Conics / Polygons	Projective invariants	Projective	Two conic correspondences; Minimal (1 pair) correspondence, approximation.
This work	Contour	Fourier Transform of sequences	Projective	Estimation of affine approximation and then projective depth iteratively for robust computation of homography.

Table 1. An Overview of Different Techniques for Homography Estimation

2. Homography from Collections

In the real world there exist many objects with sharp boundaries. These boundaries have been traditionally utilized in the form of lines, points, conics and contours, to estimate various multiview relationships. Traditionally higher order primitives such as lines and curves have been found to be more robust to track compared to points. Geometric computations, like estimation of homography or fundamental matrix, are often done robustly based on these features.

Homographies have been popular in literature for various image and video analysis tasks. Tasks like image registration have been conventionally formulated as an estimation of a similarity transform relating the points in two images. These methods were primarily based on correlation using spatial or frequency domain techniques. With the popularity of the mathematical models for imaging, homography estimation has become an integral part of applications like metric rectificationmosaicing and georeferencing. The homography between two views can be computed by finding sufficient constraints to fix eight degrees of freedom, since homographies are defined only upto scale. Homography has been estimated using many geometrical primitives. Table 1 summarizes the wide spectrum of homography estimation algorithms in a compact form. A detailed review and relative performance comparisons may be seen in [1].

The emergence of multiple view geometric techniques as a popular stream of research in computer vision [3] helped in compiling and presenting the homography estimation schemes as solution of a system of equations using Direct Linear Transformation (DLT) or other similar techniques. Algorithms such as the 4-point method based on the early known DLT [14] technique became popular. Normalization procedures were proposed as a preprocessing step to enhance numerical stability as these methods were sensitive to the exactness of the correspondence, and condition numbers of the measurement matrices. Robustness was also introduced by using standard techniques like Maximum Likelihood Estimates and RANSAC [2, 4]. These statistical techniques enhanced the robustness of these algorithm against noise in image correspondences, and hence proved to be very effective. The correspondence information used in DLT or RANSAC based homography estimation were primarily point or line correspondences.

Techniques using other geometrical primitives such as conics [8, 13] have also been developed. Polygons were used by Kumar *et al.* [6] to solve the homography estimation problem. This technique uses projective invariants like cross ratio to grow a polygon approximating a contour, present in both input images, starting with one seed point correspondence as input. This polygonal approximation is later used to estimate the homography [7].

Contours being omnipresent and general, can be tracked more easily and hence are well suited for estimating multi-view relations. A contour based technique would be robust to sensor errors and other noise as well since it is based on the properties of a collection of points and not dependent on a single point. Contours as a geometrical primitive were shown to be effective in developing rank constraints of a matrix of Fourier coefficients. The Fourier descriptors represent the shape in different views related by affine homographies [6, 10]. This technique is built on the Fourier domain based representation of contours to establish affine invariants which were shown to be helpful in solving planar shape recognition.

Texture as a geometrical primitive, has been used in literature [5, 9] for computation of affine homographies. These methods make use of the properties of the Fourier transform of corresponding textures. In [9], line patterns in textures have been used to estimate an affine approximation of the homography. This estimate is then used as an initial estimate for a nonlinear optimization procedure to determine the projective homography.

3. Fourier Domain Approach to Homography Estimation

Suppose there are N coplanar world points forming a contour. These points are imaged in two different views to form image points $\mathbf{x}[i]$ and $\mathbf{y}[i]$ ($1 \leq i \leq N$), where each $\mathbf{x}[i]$ and $\mathbf{y}[i]$ is a 2×1 vector. Assume that the homography relating the two images is $\mathbf{H}$. This can be written as

$$\begin{bmatrix} \mathbf{y}[i] \\ 1 \end{bmatrix} = \mathbf{H} \begin{bmatrix} \mathbf{x}[i] \\ 1 \end{bmatrix}. \tag{2}$$

A solution for homography estimation based on Equation 2 would assume correspondence of $\mathbf{y}[i]$ and $\mathbf{x}[i]$. When point-to-point correspondence information is not available, as is often the case, it is observed that the two sequence of points are shifted versions of each other, and related by an unknown shift λ as follows

$$\begin{bmatrix} \mathbf{y}[i] \\ 1 \end{bmatrix} = \mathbf{H} \begin{bmatrix} \mathbf{x}[i+\lambda] \\ 1 \end{bmatrix}. \tag{3}$$

The homography estimation problem can now be posed as follows: *Given N coplanar world points forming an ordered sequence, captured by two cameras, estimate the unknown homography $\mathbf{H}$ relating them.* If λ is known, this problem reduces to the homography computation from correspondence of N points. A standard point based techniques like RANSAC could be employed for this. However when only contour correspondences are available, this shift (λ) is unknown (we describe a method to estimate λ later) and attempt to solve the problem in a Fourier domain.

If the homography relating images is affine (i.e the last row of $\mathbf{H}$ is $[0\ 0\ 1]$), it follows from Equation 3 that

$$\mathbf{y}[i] = \begin{bmatrix} u_y[i] \\ v_y[i] \end{bmatrix} = \mathbf{A}\mathbf{x}[i+\lambda] + \mathbf{b}, \tag{4}$$

where $\mathbf{A}, \mathbf{b}$ are the upper-left 2×2 matrix and upper-right 2×1 vector of $\mathbf{H}$ respectively. Kuthirummal *et al.* [6, 10] have developed theory of Fourier transforms of contours in presence of affine homography. They used it effectively in solving shape recognition problem, and for estimation of affine homography. In their work algebraic constraints across multiple views based on the rank of a matrix of Fourier domain descriptor coefficients of the planar contour were derived. Kuthirummal *et al.* [6, 10] denote the Fourier domain representation of the sequence $\mathbf{y}[i]$ and $\mathbf{x}[i]$ as

$$\mathbf{Y}[k] = \begin{bmatrix} U_y[k] \\ V_y[k] \end{bmatrix}, \mathbf{X}[k] = \begin{bmatrix} U_x[k] \\ V_x[k] \end{bmatrix}, \tag{5}$$

where $\mathbf{U}_y[k]$, $\mathbf{V}_y[k]$, $\mathbf{U}_x[k]$ and $\mathbf{V}_x[k]$ are respectively the Fourier transforms of the sequences $\mathbf{u}_y[i]$, $\mathbf{v}_y[i]$, $\mathbf{u}_x[i]$ and $\mathbf{v}_x[i]$ respectively. Affine homography estimation involves computation of $\mathbf{A}$ and $\mathbf{b}$ from the point sequence correspondences.

3.1. Affine homography estimation

The translation vector $\mathbf{b}$ corresponds to the DC component ($k = 0$) in the frequency domain. It can be neglected initially by shifting the origin to the centroid of the contour. Later on $\mathbf{b}$ can be trivially computed as the difference in the centroid of the two sequences.

Since $\mathbf{H}$ is a linear transformation, it can be shown that same transformation ($\mathbf{H}$) relates the sequences in both the spatial as well as frequency domains [10]. Hence it follows that

$$\mathbf{Y}[k] = \mathbf{A}\mathbf{X}[k]e^{\frac{j2\pi\lambda k}{N}}, (k \geq 1). \tag{6}$$

Substituting Equation 5 in 6 gives,

$$\begin{bmatrix} U_y[k] \\ V_y[k] \end{bmatrix} = \mathbf{A} \begin{bmatrix} U_x[k] \\ V_x[k] \end{bmatrix} e^{\frac{j2\pi\lambda k}{N}}, \tag{7}$$

and by rearrangement, Equation 7 becomes

$$\frac{U_y[k]}{V_y[k]} = \frac{H_{11}U_x[k] + H_{12}V_x[k]}{H_{21}U_x[k] + H_{22}V_x[k]}, \tag{8}$$

where H_{ij} is (i, j)th element of $\mathbf{H}$. This is a simple linear system of equations with $2N-2$ equations and four unknowns (elements of $\mathbf{A}$). It can be solved for $\mathbf{A}$. The affine part ($\mathbf{A}$) of the homography $\mathbf{H}$ is computed like this. However this estimate is correct only upto an unknown scale factor. This scale factor can be computed by taking the ratio of the average distance of points from the centroid in the actual sequence $\mathbf{y}[i]$ and similar value in the sequence calculated by projecting $\mathbf{x}[i]$ with the current estimate of $\mathbf{A}$.

4. Iterative Projective Homography Estimation

In this process of estimating the affine homography, the strong ordering information present in the points on the contour has been utilized by transforming the sequence to the Fourier domain. The procedure of estimating affine homography from contour correspondence is not directly applicable to projective transformation. This is primarily because of the fact that the homography relationship described in Equation 1 is defined only upto scale. We can not extend the same relationship to Fourier domain. Let the projective homography is represented as $\mathbf{H} = [\mathbf{h}_1\ \mathbf{h}_2\ \mathbf{h}_3]^T$, where $\mathbf{h_i}^T$ is the ith row of $\mathbf{H}$. We can rewrite the Equation 1 as

$$\mathbf{y} = \begin{bmatrix} \mathbf{h_1}^T \\ \mathbf{h_2}^T \\ \mathbf{h_3}^T \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{h_1}^T\mathbf{x}/\mathbf{h_3}^T\mathbf{x} \\ \mathbf{h_2}^T\mathbf{x}/\mathbf{h_3}^T\mathbf{x} \\ 1 \end{bmatrix} \quad (9)$$

For a sequence of observations, it follows from Equations 2 and 9 that

$$\mathbf{y}[i] = \begin{bmatrix} u_y[i] \\ v_y[i] \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{h_1}^T\mathbf{x}[i+\lambda]/\mathbf{h_3}^T\mathbf{x}[i+\lambda] \\ \mathbf{h_2}^T\mathbf{x}[i+\lambda]/\mathbf{h_3}^T\mathbf{x}[i+\lambda] \\ 1 \end{bmatrix}, \quad (10)$$

Letting $z[i] = \mathbf{h_3}^T\mathbf{x}[i]$, Equation 10 can be rewritten as

$$z[i+\lambda]\mathbf{y}[i] = \mathbf{A}\mathbf{x}[i+\lambda] + \mathbf{b}. \quad (11)$$

This is similar to the affine relationship given in Equation 4. Here $z[i]$ is similar to the projective depth in [16]. Since $z[i]$ is unknown the affine homography estimation technique introduced in Section 3.1 can not be directly extended to estimate the projective homographies.

Claim 1 If $z[i]$s are known the terms $z[i+\lambda]\mathbf{y}[i]$ in Equation 11 can be considered as a new sequence. This new relation can then be solved for $\mathbf{A}$ and $\mathbf{b}$ (by the affine homography estimation algorithm described in the last section).

Claim 2 If affine components of the homography is known *a priori*, i.e $\mathbf{A}$ and $\mathbf{b}$ are known, $z[i]$ can be calculated from Equation 11. It can be seen that Equation 11 gives two equations for each value of i. Either of these two can be used to solve for $z[i]$.

However in practical situations the prior knowledge of neither $z[i]$ nor $\mathbf{A}$ and $\mathbf{b}$ can be assumed. Hence a two-step iterative solution is proposed to solve the projective homography estimation problem. This is in line with the general theory of function optimization of multiple variables where a function is optimized with respect to a set of variables in one step and then with respect to the other variables in another step.

- **Step 1**: Assuming some values for projective depth estimate the affine components.
- **Step 2**: Estimate the projective depth from the current estimate of affine components.

Claim 3 Re-projection error is the mean squared error between original sequence $\mathbf{y}[i]$ and transformed sequence $\mathbf{H}_{est}\mathbf{x}[i+\lambda]$, where $\mathbf{H}_{est}$ is the current estimate of homography. The observation is that the re-projection error decreases with each iteration. In Step 1, the re-projection error is minimized with respect to $\mathbf{A}$ and $\mathbf{b}$. In Step 2, it is minimized with respect to an unknown $z[i]$. Hence the error decreases with each iteration.

Claim 4 Re-projection error is by definition non-negative. When true values for $\mathbf{A}$, $\mathbf{b}$ and $z[i]$ are achieved, the re-projection error becomes zero. This follows directly from Equation 11 by substituting the true values for $\mathbf{A}$, $\mathbf{b}$ and $z[i]$. In a synthetic situation (where the coordinates are real numbers) it is possible to achieve a zero error. However this value would be only close to zero in a practical situation where discretization and other errors are present.

4.1. Estimation of $\mathbf{h_3}$

Affine homography algorithm explained in section 3.1 is used in Step 1 of the two-step iterative homography estimation scheme. Step 2 of the algorithm is now explained in detail. Note that since $z[i] = \mathbf{h_3}^T\mathbf{x}[i]$, $z[i]$ is easily estimated once there is an estimate for $\mathbf{h_3}$. A procedure to estimate $\mathbf{h_3}$ is explained below. It is assumed that λ is known at this stage. Later this assumption is relaxed by introducing an estimation technique for λ.

A general projective matrix $\mathbf{H}$ is a full-rank matrix. In other words, $\mathbf{h_1}$, $\mathbf{h_2}$ and $\mathbf{h_3}$ are linearly independent. To estimate $\mathbf{h_3}$, an expression for $\mathbf{h_3}$ is needed. Such an expression should be expressed in terms of both $\mathbf{h_1}$ and $\mathbf{h_2}$. Noting that $\mathbf{h_i}$s are vectors in $\mathcal{R}^3$, an appropriate set of bases of $\mathcal{R}^3$ which can express any vector $\mathbf{h_3}$ would be $\{\mathbf{h_1}, \mathbf{h_2}, \mathbf{h_1} \times \mathbf{h_2}\}$. In other words, $\mathbf{h_3}$ can be expressed as a linear combination of the above bases,

$$\mathbf{h_3} = \alpha\mathbf{h_1} + \beta\mathbf{h_2} + \gamma(\mathbf{h_1} \times \mathbf{h_2}). \quad (12)$$

The problem has now reduced to one of estimating appropriate values of α, β and γ. Rearranging Equation 10, to get

$$\mathbf{a}\alpha + \mathbf{b}\beta + \mathbf{c}\gamma + \mathbf{d} = \mathbf{0}, \tag{13}$$

where,

$$\begin{aligned}
\mathbf{a} &= u_y[i](\mathbf{h_1}^T\mathbf{x}[i+\lambda]) \\
\mathbf{b} &= u_y[i](\mathbf{h_2}^T\mathbf{x}[i+\lambda]) \\
\mathbf{c} &= u_y[i]((\mathbf{h_1} \times \mathbf{h_2})^T\mathbf{x}[i+\lambda]) \\
\mathbf{d} &= -\mathbf{h_1}^T\mathbf{x}[i+\lambda].
\end{aligned}$$

There are similar equations with terms of $v_y[i]$. Since there are $2N$ equations for various values of i in Equation 13 and three unknowns, α, β and γ can be calculated. Then $\mathbf{h}_3$ can be computed from Equation 12.

4.2. Estimation of shift λ

Equation 7 after rearrangement gives

$$\frac{U_y[k]}{H_{11}U_x[k] + H_{12}V_x[k]} = e^{\frac{j2\pi\lambda k}{N}}. \tag{14}$$

Similar equation can be written with $V_y[k]$. By the translation property of Fourier Transforms it can be shown that the inverse Fourier transform of the sequence of terms in Equation 14 would be an impulse shifted by λ. The location of this impulse in the spatial domain would indicate the value of λ. [10] gives another method to estimate λ when the homography is assumed to be projective.

4.3. Algorithm

The complete algorithm is as follows:

Input: $x[i], y[i]$ with N points each, where $1 < i \leq N$
Output: Homography matrix $\mathbf{H}$

Initialize $z[i] = 1, \forall i$.
repeat
  Solve for $\mathbf{h_1}$ and $\mathbf{h_2}$ using current values for $\mathbf{y}[i]$, $\mathbf{x}[i+\lambda]$ and $z[i]$;
  Solve for λ;
  Estimate $\mathbf{h_3}$ using the updated values of $\mathbf{h_1}$, $\mathbf{h_2}$ and λ;
  Update $z[i] = \mathbf{h_3}^T\mathbf{x}[i]$;
until convergence.

The homography estimated by the algorithm presented above would depend on the initialization values for the projective depth. It is found that initializing projective depth $z[i] = 1 \; \forall \; i$ proves to be a suitable initialization. This effectively amounts to starting with a weak perspective projection model. The convergence condition should be chosen carefully. One possibility is to chose the Mean Square Error (MSE) between the second sequence and the sequence obtained by projecting first sequence with the current estimate of $\mathbf{H}$, i.e.,

$$MSE = \frac{1}{N}\sum_i(\mathbf{y}[i] - \mathbf{H}\mathbf{x}[i+\lambda])^2.$$

It can be shown that the proposed algorithm minimizes MSE in both steps and hence converges to the correct solution.

5. Results and Discussions

The proposed method was tested extensively in a range of experiments. Some of the contours used for these experiments are shown in Figure 1. These contours were chosen after careful examination. The criteria used in selecting these images include – the number of points, a rich variety of curvature patterns, contours encountered in real-life, complexity etc.

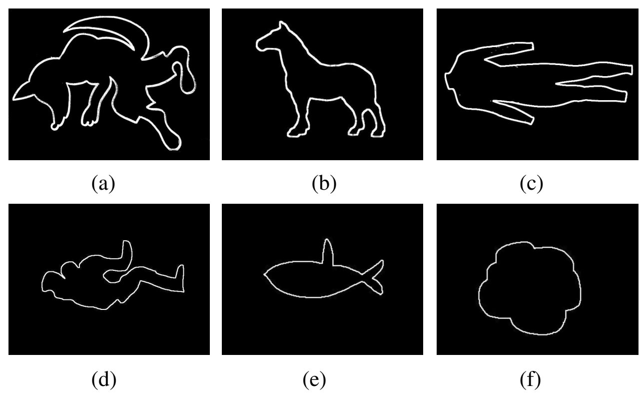

Figure 1. Figure shows various contours used in the experiments. The contours vary in their shape, curvature properties, number of points etc.

In the first experiment we apply a range of ground truth homographies and measure the correctness of the estimated solution. The generation of these homographies can be explained by considering a virtual camera. Assume that a virtual camera is moved around a scene plane (which coincides with the $z = 0$ plane). Consider that camera is rotated from $-60°$ to $+60°$ each around both x and y axis, with fixed radius. Consider the image seen by the camera placed at (θ, ϕ) angles about x and y axes. There is a homography between the image seen and an image taken when the camera is placed at $(0, 0)$. This homography became one of the homographies considered for our experiments. Many such homographies can be generated by changing the values of

(θ, ϕ). The advantage with this scheme of generating a homography is that it covers the most realistic poses at which, actual images are generally taken.

We consider 10000 such equally sampled homographies and apply them over each of the contours shown in Figure 1 and generat a new view of the given contour for each contour. These two contours are used as input to our algorithm and we estimate the homography. The algorithm was found to converge to the correct solution in each case within very few iterations (typically 3 to 5). The re-projection error (error between the actual second contour and the warped contour generated by applying the estimated homography over the initial contour) decreased very sharply as shown in a typical example in Figure 2. Homographies beyond this range (i.e $\pm 60°$) show some errors in the estimate and some examples of such contours are shown in Figure 5. The images in this range are highly distorted and it is difficult for even a human to identify the contours.

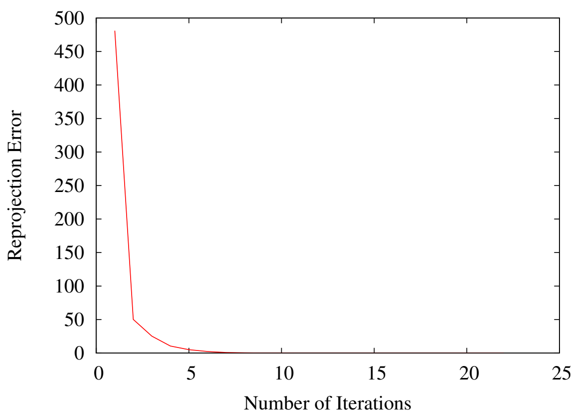

Figure 2. A typical plot of re-projection error over number of iterations. The error decreases rapidly with iterations.

In another experiment we consider a variety of images from various situations in real-life. These images are used to demonstrate the effectiveness of our algorithm in a variety of real-life situations. Some of the images are shown in Figure 3. The figure is arranged as follows. In parts (a) and (b) we list the input images. Parts (c) and (d) illustrate the contours extracted from these images. (e) shows the overlay of contour (a) transformed by estimated homography over (b). Figure 4 shows tux example in greater detail. The high overlap between the contours clearly shows the correctness of our algorithm. This experiment shows that the given algorithm converges to the correct homography for a variety of situations. This makes the algorithm acceptable for various real-life situations.

Parameters such as the number of points, noise (dis-

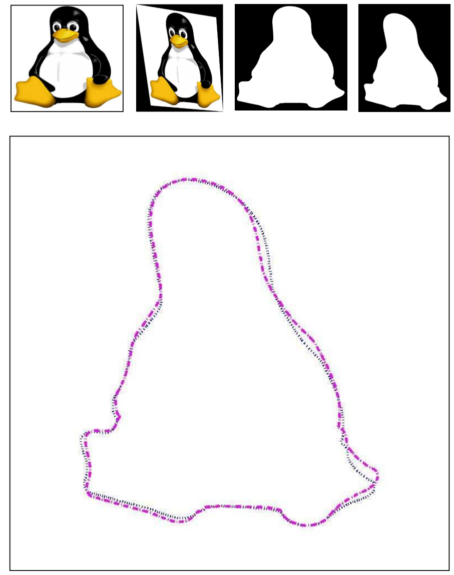

Figure 4. Overlay of first tux contour wrapped over second contour by the estimated homography. The magenta colored dash-dotted line is the warped contour and blue colored dotted line is the actual contour. The original images and their contours are also shown above.

cretization, sampling, localization etc.), symmetry in contour, occlusion, projectiveness of homography etc. can affect the performance of the proposed algorithm. We present the analysis of proposed algorithm with respect to these parameters.

We measure the performance of our algorithm with noise. For this we add Gaussian noise with standard deviation $\sigma \in \{2, 3, \ldots, 8\}$ pixels along each coordinate to $\alpha \in \{5, 10, 15, 20\}$ % points. We perform the experiment on the objects of Figure 1 with the ground truth homographies generated as before. It was noted that the successful convergence depended on the severity of noise and also on the complexity of contours. Simpler contours tended to have good estimates even in presence of severe noise. This may be attributed to the fact that simpler contours preserved their inherent structure better than complex contours in the presence of structure distorting noise. The results are tabulated in Table 2.

Symmetry in contour introduces ambiguity in the homography estimation. This is so because different homographies map a view of a symmetric object to the same target view because of the symmetry. Hence all homography estimation algorithms suffer from such symmetrical nature of the target scene. If a contour is symmetrical the constituent signal of its x or y components (i.e. all the x or y coordi-

1) 2) 3)

(a) (b) (c) (d) (e)

Figure 3. (a), (b) Input images. (c), (d) Contours extracted from (a) and (b) respectively, (e) Overlay of contour (d) with (c) warped by the estimated homography. (1) shows a satellite image of a lake, (2) shows an outdoor advertisement, (3) shows a part of a name board.

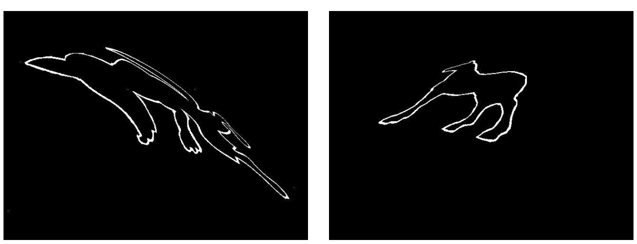

Figure 5. Examples of failure cases of the proposed technique. The left and right image are respectively highly deformed images of Figure 1 (a) and (b)

σ/α	5 %	10 %	15 %	20 %
2	100	100	100	100
3	100	100	100	100
4	100	100	98.7	96.3
5	100	99.1	98.4	94.5
6	96.3	95.5	94.3	93.7

Table 2. Performance of the proposed scheme in the presence of Gaussian noise.

nates of the contour sequence taken together) might be periodic in nature. This results in a poorer performance of our algorithm due to lack of sufficient frequency components. However to overcome this lack of frequency components we can analyze the Fourier representation of the contours for the presence of few dominant frequencies. If such is the case then one can find out the time period of the contour's repeating pattern. Using this information we can extract the non repeating part of the contour in the two images and use them to estimate the homography.

In another experiment we measure the effect of the number of points in the contour on the successful convergence of the estimation procedure or of our algorithm. We consider the same contours as before and sub-sampled them at various intervals till a minimum of 30 points in each contour, to generate a large number of test cases. We generated 10000 homographies as before. After estimating the homography for each sub-sampling, we found that the method performed reasonably well for larger number of points, with gradual degradation in performance as the number of points decreases. The algorithm is unsuitable for situations where there are only few points (typically < 30) in the contour. In most applications, the number of points in the contours is well within this range for our algorithm to converge successfully.

In another experiment, we measure the variation of performance as the *projectiveness* of the applied homography is varied. For this we define projectivity p as $p = |\mathbf{h}_3|/|\mathbf{H}|$. This measure captures the projective component after normalizing with respect to the affine components. In this experiment we use the same setup as in first experiment while recording the projectivity for each homography. We observed that while the value of projectivity is within an acceptable range (< 0.6), the algorithm converges to the correct solution. However with greater projectivity, the performance of our algorithm decreases slightly. We found that in most practical applications the value of projectivity for homographies is lower than 0.6. For same projectivity other affine approximation based approaches give poorer performance than our approach.

6. Conclusion and Future Work

We have proposed a homography estimation technique that uses correspondence of contours. Our technique does not need explicit point-to-point correspondence. In fact this point-to-point correspondence is obtained as a by-product of the homography computation procedure. We have demonstrated the applicability of our technique to a variety of real world problems. Future work would include applying Fourier domain based representation for projective homography estimation from texture regions and wide baseline stereo.

Acknowledgments

The authors would like to acknowledge the help given by Sujit Kuthirummal in reviewing the drafts and with the code of his earlier works.

References

[1] A. Agarwal, C. V. Jawahar, and P. J. Narayanan. A Survey of Planar Homography Estimation Techniques. IIIT Technical Report IIIT/TR/2005/12, June 2005.

[2] M. A. Fischler and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of Association of Computing Machines*, 24:381–395, June 1981.

[3] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision, 2nd Edition*. Cambridge University Press, 2004.

[4] K. Kanatani. *Statistical Optimization for Geometric Computation: Theory and Practice*. Elsevier Science, 1996.

[5] S. Kruger and A. Calway. Image Registration Using Multiresolution Frequency Domain Correlation. In *British Machine Vision Conference (BMVC)*, pages 316–325, September 1998.

[6] M. P. Kumar, S. Goyal, S. Kuthirummal, C. V. Jawahar, and P. J. Narayanan. Discrete Contours in Multiple Views: Approximation and Recognition. *Journal of Image and Vision Computing (IVC)*, 22(14):1229–1239, December 2004.

[7] M. P. Kumar, C. V. Jawahar, and P. J. Narayanan. Building Blocks for Autonomous Navigation Using Contour Correspondences. In *Proceedings of the International Conference on Image Processing (ICIP)*, October 2004.

[8] M. P. Kumar, C. V. Jawahar, and P. J. Narayanan. Geometric Structure Computation from Conics. In *Proceedings of Indian Conference on Computer Vision, Graphics, and Image Processing (ICVGIP)*, pages 9–14, 2004.

[9] M. P. Kumar, S. Kuthirummal, C. V. Jawahar, and P. J. Narayanan. Planar Homography from Fourier Domain Representation. In *Proceedings of the International Conference on Signal Processing and Communications (SPCOM)*, December 2004.

[10] S. Kuthirummal, C. V. Jawahar, and P. J. Narayanan. Planar Shape Recognition across Multiple Views. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 482–488, August 2002.

[11] Q. Luong and T. Vieville. Canonical Representation for the Geometries of Multiple Views. *Computer Vision and Image Understanding (CVIU)*, 64(2):193–229, 1996.

[12] H. Y. Shum and R. Szeliski. Construction of Panoramic Image Mosaics with Global and Local Alignment. *International Journal of Computer Vision (IJCV)*, 36(2):101–130, 2000.

[13] A. Sugimoto. A Linear Algorithm for Computing the Homography from Conics in Correspondence. *Journal of Mathematical Imaging and Vision*, 13:115–130, 2000.

[14] I. E. Sutherland. A Man-Machine Graphical Communication System. Technical Report 296, MIT Lincoln Laboratories, 1963.

[15] R. Szeliski and H. Y. Shum. Creating Full View Panoramic Image Mosaics and Environment Maps. In *Proceedings of Computer Graphics (SIGGRAPH)*, pages 251–258, 1997.

[16] B. Triggs. Factorization Methods for Projective Structure and Motion. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 845–51, 1996.

[17] Z. Zhang. A Flexible New Technique for Camera Calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 22:1330–1334, November 2000.