A Survey of Planar Homography Estimation Techniques

Anubhav Agarwal, C. V. Jawahar, and P. J. Narayanan

Centre for Visual Information Technology
International Institute of Information Technology
Hyderabad 500019 INDIA Email: pjn@iiit.ac.in

Abstract

The estimation of the homography between two views is a key step in many applications involving multiple view geometry. The homography exists between two views between projections of points on a 3D plane. A homography exists also between projections of all points if the cameras have purely rotational motion. A number of algorithms have been proposed for the estimation of the homography relation between two images of a planar scene. They use features or primitives ranging from simple points to a complex ones like non-parametric curves. Different algorithms make different assumptions on the imaging setup and what is known about them. This article surveys several homography estimation techniques from the literature. The essential theory behind each method is presented briefly and compared with the others. Experiments aimed at providing a representative analysis and comparison of the methods discussed are also presented in the paper.

1 Introduction

The planar homography is a non-singular linear relationship between points on planes. The homography between two views plays an important role in the geometry of multiple views [4, 5, 8]. Images of points on a plane in one view are related to corresponding image points in another view by a planar homography using a homogeneous representation. This is a projective relation since it only depends on the intersection of planes with lines. The homography transfers points from one view to the other as if they were images of points on the plane. The homography induced by a plane is unique up to a scale factor and is determined by 8 parameters or degrees of freedom. The homography depends on the intrinsic and extrinsic parameters of the cameras used for the two views and the parameters of the 3D plane.

Computing the homography induced by a plane between two views is an important step towards calibration [34], metric rectification [3, 20], reconstruction, and other applications that use perspective geometry such as image registration and image mosaicing [30, 33]. Homography is also used in tracking applications using multiple cameras and to build projector-camera systems [2]. We can use the homography relation between two views to infer the transformation between planes. Even when the target is partially or fully occluded by an unknown object, the tracker can follow the target as long as it is visible from another view. No complicated inference scheme is used to fuse the multiple camera observation and no 3D information is recovered explicitly. Metric rectification is the task of obtaining the fronto-parallel view of an object given a projectively distorted image of it. Approaches to metric rectification make use of parallel and perpendicular lines in the planar object, often under human intervention. There are many situations where the known, undistorted shape of a feature in the image can be used for the estimation of the homography matrix. The transformation can then be used for the fronto-parallel view of the image.

Planar homography between two views can be determined by finding sufficient constraints to fix the (up to) 8 degrees of freedom of the relation. Homography can be estimated from the matching of 4 points or lines or their combinations in general positions in two views. Each matching pair gives two constraints and fixes two degrees of freedom. In practice, robust statistical techniques are employed on a large number of matching points or lines after normalizing the data to reduce the adverse effects of noise, quantization, etc [11]. The degrees of freedom can be fixed by matching other parametric and non-parametric curves or contours in the images. Other gross properties in the image such as texture can also be used to compute the planar homography between two views [17, 22]. A homography relates two views of a scene when the transformation between the two cameras is purely rotational about an axis through its principal point [8]. Such homographies can also be estimated using corresponding features as they provide similar constraints as planar homographies.

In this paper we present a survey of the different homography estimation techniques reported in the literature. We present the essential theory behind each method and the briefly explain how each is different from the others. We also present experimental comparison of these techniques on a number synthetic images under different conditions. In Section 2, we explain the notation and background material on homography including parameterizations useful for its computation. In Section 3, we present the different estimation techniques. The experimental comparison of the techniques is given in Section 4. Section 5 presents discussions and conclusions of the survey.

2 Planar Homography: Basics

We present the notation used in this paper and the basics of multiview imaging and the associated geometric properties in this section.

2.1 Notation

We use lowercase boldface letters like $\mathbf{x}$ for vectors and boldface capital letters like $\mathbf{A}$ for matrices. $\mathbf{x^T}$ is the transpose of vector $\mathbf{x}$. The cross-product of two vectors $\mathbf{x}$ and $\mathbf{y}$ is represented by $\mathbf{x} \times \mathbf{y}$. The antisymmetric matrix such that $[\mathbf{v}]_\times \mathbf{x} = \mathbf{v} \times \mathbf{x}$ for all vectors $\mathbf{x}$ is noted $[\mathbf{v}]_\times$. Lines and points are represented using homogeneous vectors. A point (x, y) in Cartesian coordinates is represented as a homogeneous vector $\mathbf{x} = [x, y, 1]^T$. A line in the plane is represented in coordinate geometry by the equation $ax + by + c = 0$; different choices of a, b and c giving rise to different lines. A line can be represented by the homogeneous vector $\mathbf{l} = [a, b, c]^T$. A point $\mathbf{x} = (x, y)$ lies on the line $\mathbf{l}$ if and only if the inner product of $\mathbf{l}$ and $\mathbf{x}$ vanishes, i.e., $\mathbf{l} \cdot \mathbf{x} = \mathbf{l^T x} = \mathbf{0}$. Similarly, points and planes in 3D space are represented using homogeneous 4-vectors $\mathbf{X}$ and π respectively. Incidence of a point on the plans is given by $\pi^T \mathbf{X} = \mathbf{0}$.

A conic is a curve represented by a second-degree equation in the plane. The equation of a conic in inhomogeneous coordinate is a polynomial of degree 2 given by $ax^2 + bxy + cy^2 + dx + ey + f = 0$. Homogenizing this by the replacements: $x \to x_1/x_3, y \to x_2/x_3$ gives $ax_1^2 + bx_1x_2 + cx_2^2 + dx_1x_3 + ex_2x_3 + fx_3^2 = 0$, or in matrix form

$$\mathbf{x^T C x} = 0, \quad \text{where} \quad \mathbf{C} = \begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix}. \tag{1}$$

The matrix $\mathbf{C}$, called the conic coefficient matrix, is symmetric and gives the homogeneous representation of a conic.

A general algebraic curve of order n admits a tensor representation: $\mathbf{T}_{i_1 \ldots i_n} \mathbf{p}^{i_1} \ldots \mathbf{p}^{i_n} = 0$ for a point $\mathbf{p}$ belonging to the curve, where for each k, $i_k \in 1, 2, 3$ for x, y and z coordinates. Thus, for a conic, $n = 2$ and we get a 2-dimensional matrix similar to Equation 1.

2.2 The projective camera model

The pin-hole camera model is most widely used and performs a perfect perspective transformation of 3D space on to a retinal plane. In the general case, we must also account for a change of world coordinates, as well as for a change of retinal coordinates, so that a generalization of the previous assumption is that the camera performs a projective linear transformation rather than a mere perspective transformation. The pixel coordinates u and v are the only information we have if the camera is not calibrated. In a calibrated camera, the relation between the world point $\mathbf{M}$ with coordinates (X, Y, Z) and its projection in the image $\mathbf{m}$ with coordinates (u, v) is given by

$$\mathbf{m} = \begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \mathbf{K} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{G} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \equiv \mathbf{PM},$$

where $\mathbf{K}$ is a 3×3 intrinsic matrix accounting for camera sampling and optical characteristics and $\mathbf{G}$ is a 4×4 displacement matrix accounting for camera position and orientation in the world coordinate frame.

Observe that $\mathbf{PM} = (\mathbf{PH})(\mathbf{H^{-1}M})$ and similarly for $\mathbf{P}'$. Thus, if $\mathbf{x} \leftrightarrow \mathbf{x}'$ are matched points with respect to the pair of cameras $(\mathbf{P}, \mathbf{P}')$, corresponding to a 3D point $\mathbf{M}$, then they are also matched points with respect to the pair of cameras $(\mathbf{PH}, \mathbf{P}'\mathbf{H})$, corresponding to the point $\mathbf{H^{-1}M}$. Given this ambiguity, it is common to define a specific canonical form for the pair of camera matrices in which the first matrix is of the simple form $[\mathbf{I} \mid \mathbf{0}]$. To see that this is always possible, let $\mathbf{P}$ be augmented by one row to make a non-singular matrix, denoted $\mathbf{P}^*$. Now, letting $\mathbf{H} = \mathbf{P}^{*-1}$, one verifies that $\mathbf{PH} = [\mathbf{I} \mid \mathbf{0}]$ as desired.

2.3 Homography induced by a plane

When a planar object is imaged from multiple view-points or when a scene is imaged by cameras having the same optical centre, the images are related by a unique homography. For a plane $\pi = [\mathbf{v}^T, 1]^T$ for a vector $\mathbf{v}$ in the scene, the ray corresponding to a point $\mathbf{x}$ meets it at a point $\mathbf{X}_\pi$, which projects to $\mathbf{x}'$ in the other image. Given the projection matrices $\mathbf{P} = [\mathbf{I} \mid \mathbf{0}]$ and $\mathbf{P}' = [\mathbf{A} \mid \mathbf{a}]$ for the two views, the homography induced by the plane is given by (assuming $\pi_4 = 1$ since the plane does not pass through the centre of the first camera) [8]

$$\mathbf{x}' = \mathbf{Hx} \quad \text{with} \quad \mathbf{H} = \mathbf{A} - \mathbf{av^T}. \tag{2}$$

If the cameras have different intrinsic matrices $\mathbf{K}'$ and $\mathbf{K}$ respectively, the homography due the plane is given by

$$\mathbf{H} = \mathbf{K}'(\mathbf{A} - \mathbf{av^T})\mathbf{K^{-1}}. \tag{3}$$

2.4 Rotational Homography

A similar homography relation maps all corresponding points in two views when the cameras share their principal points. The transformation between the cameras in that case is purely rotational. The two views are two planar

cross-sections of the plenoptic function [1] or the field of rays through the principal point. Each cross-section is a perspectivity and the transformation between them is a general projectivity or a projective homography [29]. The conjugate rotation homography for pure rotation $\mathbf{R}$ between the cameras is given by [8]

$$\mathbf{H} = \mathbf{K}'\mathbf{R}\mathbf{K}^{-1}. \tag{4}$$

This formula also gives the homography induced by the plane at infinity for any pair of cameras with the same rotation.

2.5 Transferring points, lines, and conics

A homography $\mathbf{H}$ can be used to transfer other feature types on the plane from one view to the other. A point $\mathbf{x}$ on the plane can be transferred to its image $\mathbf{x}'$ in the other view using

$$\mathbf{x}' = \mathbf{H}\mathbf{x}. \tag{5}$$

Similary, a line $\mathbf{l}$ may be transferred to its corresponding line $\mathbf{l}'$ in the other view using

$$\mathbf{l}' = \mathbf{H}^{-\mathbf{T}}\mathbf{l}. \tag{6}$$

Corresponding conics are related by

$$\mathbf{C}' = \mathbf{H}^{-\mathbf{T}}\mathbf{C}\mathbf{H}^{-\mathbf{1}}. \tag{7}$$

Features like contours or parametric curves may not be transferred directly using the homography. However, the above relations hold between any pair of matching points, lines, and conics. Points on an arbitrary contour obey Equation 5 even if there is no method to transfer the contour as a whole. Similarly, the sides of a polygon will obey Equation 6 even when there is no way to directly transfer a polygon or a rectangle using the homography.

2.6 The Fundamental Matrix

The Fundamental Matrix $\mathbf{F}$ for two images acquired by cameras with non-coincident centres is the 3×3, rank 2 homogeneous matrix which satisfies

$$\mathbf{x}'^{\mathbf{T}}\mathbf{F}\mathbf{x} = \mathbf{0}, \tag{8}$$

for all corresponding points $\mathbf{x} \leftrightarrow \mathbf{x}'$, i.e., $\mathbf{x}$ and $\mathbf{x}'$ are the projections into the two cameras of the same world point. The fundamental matrix is not defined for cameras that share the principal point. If the canonical cameras are $\mathbf{P} = [\mathbf{I} \mid \mathbf{0}]$ and $\mathbf{P}' = [\mathbf{M} \mid \mathbf{m}]$m, then the fundamental matrix $\mathbf{F}$ is given by

$$\mathbf{F} = [\mathbf{e}']_{\times}\mathbf{M}, \tag{9}$$

where $\mathbf{e}' = \mathbf{m}$ and $\mathbf{e} = \mathbf{M}^{-\mathbf{1}}\mathbf{m}$. The points $\mathbf{e}$ and $\mathbf{e}'$ are called the epipoles, which are the points of intersection of the line joining the camera centers with image planes. Equivalently, it is the image in one view of the other camera center. Further, $\mathbf{e}$ and $\mathbf{e}'$ are the right and left null-space of $\mathbf{F}$ satisfying

$$\mathbf{e}'^{\mathbf{T}}\mathbf{F} = \mathbf{0} \quad \text{and} \quad \mathbf{F}\mathbf{e} = \mathbf{0}. \tag{10}$$

Luong and Faugeras present a detailed survey of the different techniques used in computing the Fundamental Matrix [23].

2.7 Planar homography and epipolar geometry

The epipolar geometry determines the projective geometry between two views, and can be used to define conditions on homographies which are induced by actual scene planes. For example, since correspondences $\mathbf{x} \leftrightarrow \mathbf{x}' = \mathbf{H}\mathbf{x}$ obey the epipolar constraint if $\mathbf{H}$ is induced by a plane, then from $\mathbf{x}'^{\mathbf{T}}\mathbf{F}\mathbf{x} = \mathbf{0}$, we get $(\mathbf{H}\mathbf{x})^{\mathbf{T}}\mathbf{F}\mathbf{x} = \mathbf{x}^{\mathbf{T}}\mathbf{H}^{\mathbf{T}}\mathbf{F}\mathbf{x} = \mathbf{0}$ for all $\mathbf{x}$. Hence, a homography $\mathbf{H}$ is compatible with a fundamental matrix if and only if the matrix $\mathbf{H}^{\mathbf{T}}\mathbf{F}$ is skew-symmetric, i.e,

$$\mathbf{H}^{\mathbf{T}}\mathbf{F} + \mathbf{F}^{\mathbf{T}}\mathbf{H} = \mathbf{0}.$$

This compatibility constraint is an implicit equation in $\mathbf{H}$ and $\mathbf{F}$. We can get an explicit expression for a homography $\mathbf{H}$ induced by a plane given $\mathbf{F}$ which is more suitable for a computational algorithm. Given the fundamental matrix between two views, the three-parameter family of homographies induced by a world plane is

$$\mathbf{H} = \mathbf{A} - \mathbf{e}'\mathbf{v}^{\mathbf{T}}, \tag{11}$$

where $[\mathbf{e}']_{\times}\mathbf{A} = \mathbf{F}$ is any decomposition of the fundamental matrix. This results from Equations 2 and 9 by choosing the two cameras to be $[\mathbf{I} \mid \mathbf{0}]$ and $[\mathbf{A} \mid \mathbf{e}']$.

2.8 Sufficient number of measurements

We look at the number of constraints in terms of corresponding features required to estimate the projective transformation $\mathbf{H}$. A lower bound is available from the number of degrees of freedom and the number of constraints. The matrix $\mathbf{H}$ contains 9 entries, but is defined only up to scale. Thus, the total number of degrees of freedom in a 2D projective transformation is 8. Each corresponding 2D point or line generates two constraints on $\mathbf{H}$ by Equation 5 and hence the correspondence of four points or four lines is sufficient to compute $\mathbf{H}$. For a planar affine transformation with 6 degrees of freedom, only three corresponding points or lines are required, and so on.

A conic provides five constraints on a 2D homography. Hence two matching pairs of conics are sufficient to recover the homography. A planar polynomial algebraic curve of order n is represented by a polynomial containing $\frac{1}{2}n(n+3)+1$ terms. Each matching curve thus provides with $\frac{1}{2}n(n+3)$ equations on the entries of homography matrix. Thus, a single matching pair of curves of order $n \geq 3$ are sufficient for uniquely recovering the homography matrix induced by the plane of curve in space.

In practice, the features like points, lines, and conics detected in the image could be noisy to get good solutions using the minimum numbers of them. A large number of features are used to make the solutions robust [8].

3 Analysis of Homography Estimation Methods

3.1 Using points and lines

Points and lines are the simplest and fundamental features that can be used for estimating homography. They have also been the main focus of the researchers for homography estimation. The main point-based and line-based methods that can been employed for estimating homography are discussed below.

The methods described in this section recover the homography as a linear relationship on point and line features. These methods can therefore be applied to recover homography induced by the plane as well as rotational homography, given sufficient number of points/lines.

3.1.1 Direct Linear Transformation (DLT) using point correspondences

We begin with a simple linear algorithm for determining $\mathbf{H}$ given a set of four 2D to 2D point correspondences, $\mathbf{x_i} \leftrightarrow \mathbf{x_i'}$. The equation $\mathbf{x'} = \mathbf{Hx}$ may be expressed in terms of the vector cross product as $\mathbf{x_i'} \times (\mathbf{Hx_i}) = \mathbf{0}$. If the j-th row of the matrix $\mathbf{H}$ is denoted by $\mathbf{h}^{j\mathbf{T}}$, we have

$$\mathbf{Hx_i} = \begin{pmatrix} \mathbf{h^{1T}x_i} \\ \mathbf{h^{2T}x_i} \\ \mathbf{h^{3T}x_i} \end{pmatrix}.$$

Writing $\mathbf{x_i'} = (x_i', y_i', w_i')^T$, the cross product can be given explicitly as

$$\mathbf{x_i'} \times \mathbf{Hx_i} = \begin{pmatrix} y_i'\mathbf{h^{3T}x_i} - w_i'\mathbf{h^{2T}x_i} \\ w_i'\mathbf{h^{1T}x_i} - x_i'\mathbf{h^{3T}x_i} \\ x_i'\mathbf{h^{2T}x_i} - y_i'\mathbf{h^{1T}x_i} \end{pmatrix}.$$

This can be rewritten in the form

$$\begin{bmatrix} \mathbf{0^T} & -w_i'\mathbf{x_i^T} & y_i'\mathbf{x_i^T} \\ w_i'\mathbf{x_i^T} & \mathbf{0^T} & -x_i'\mathbf{x_i^T} \\ -y_i\mathbf{x_i^T} & x_i'\mathbf{x_i^T} & \mathbf{0^T} \end{bmatrix} \begin{bmatrix} \mathbf{h^1} \\ \mathbf{h^2} \\ \mathbf{h^3} \end{bmatrix} = \mathbf{0}. \tag{12}$$

These equations have the form $\mathbf{L_i h} = \mathbf{0}$ where $\mathbf{L_i}$ is a 3×9 matrix, and let $\mathbf{h} = [\mathbf{h}^{1T}\ \mathbf{h}^{2T}\ \mathbf{h}^{3T}]^T$. Although Equation 12 contains three equations, only two of them are linearly independent. Thus each point correspondence gives two equations in the entries of $\mathbf{H}$. The set of equations can be written as

$$\begin{bmatrix} \mathbf{0^T} & -w_i'\mathbf{x_i^T} & y_i'\mathbf{x_i^T} \\ w_i'\mathbf{x_i^T} & \mathbf{0^T} & -x_i'\mathbf{x_i^T} \end{bmatrix} \begin{bmatrix} \mathbf{h^1} \\ \mathbf{h^2} \\ \mathbf{h^3} \end{bmatrix} = \mathbf{0}. \tag{13}$$

Given a set of four point correspondences from the plane, a set of equations $\mathbf{Lh} = \mathbf{0}$ is obtained, where $\mathbf{L}$ is the matrix obtained by stacking the rows of $\mathbf{L_i}$ contributed from each correspondence and $\mathbf{h}$ is the vector of unknown entries of $\mathbf{H}$.

In practice, the extracted image points do not satisfy the relation $\mathbf{x'} = \mathbf{Hx}$ because of noise in the extracted image points. Let us assume that $\mathbf{x_i'}$ is corrupted by Gaussian noise with mean $\mathbf{0}$ and covariance matrix $\mathbf{\Sigma_{x'}}$. Then, the maximum likelihood estimation of $\mathbf{H}$ is obtained by minimizing the following functional [32]

$$\mathbf{J} = \sum \left(\mathbf{x_i'} - \hat{\mathbf{x}}_\mathbf{i}'\right)^\mathbf{T} \mathbf{\Sigma}_{\mathbf{x'}}^{-\mathbf{1}} \left(\mathbf{x_i'} - \hat{\mathbf{x}}_\mathbf{i}'\right), \quad \text{where} \quad \hat{\mathbf{x}}_\mathbf{i}' = \frac{\mathbf{1}}{\mathbf{h^{3T}x_i}} \begin{bmatrix} \mathbf{h^{1T}x_i} \\ \mathbf{h^{2T}x_i} \end{bmatrix},$$

with $\mathbf{h^i}$ being the ith row of $\mathbf{H}$. In practice, assume $\lambda\mathbf{x_i'} = \sigma^2\mathbf{I}$ for all i. The above problem becomes a nonlinear least-squares estimation problem, i.e., finding $\mathbf{H}$ such that $\|\mathbf{x_i'} - \hat{\mathbf{x}}_\mathbf{i}'\|^2$ is minimum. Then

$$\begin{bmatrix} \mathbf{x}^T & \mathbf{0} & -u\mathbf{x}^T \\ \mathbf{0} & \mathbf{x}^T & -v\mathbf{x}^T \end{bmatrix} \mathbf{h} = 0. \tag{14}$$

Given n points, they can be written in matrix equation as $\mathbf{Lh} = \mathbf{0}$, where $\mathbf{L}$ is a $2n \times 9$ matrix. We seek a non-zero solution $\mathbf{h}$ that minimizes a suitable cost function subject to the constraint $\|\mathbf{h}\| = 1$. This is identical to the problem of finding the minimum of the quotient $\|\mathbf{Lh}\|/\|\mathbf{h}\|$. The solution is the (unit) eigenvector of $\mathbf{L^T L}$ with the least eigenvalue. Equivalently, the solution is the right singular vector associated with the smallest singular value of $\mathbf{L}$.

In $\mathbf{L}$, some elements are constant 1, some are in pixels, some are in world coordinates, and some are multiplication of both. This makes $\mathbf{L}$ poorly conditioned numerically. Much better results can be obtained by performing a simple data normalization, prior to running the above procedure. The procedure is suggested as follows.

Algorithm 1 Direct Linear Transformation on points

Goal: Given $n \geq 4$ 2D to 2D point correspondences $\mathbf{x_i} \leftrightarrow \mathbf{x'_i}$, determine the 2D homography matrix $\mathbf{H}$ such that $\mathbf{x'_i} = \mathbf{H}\mathbf{x_i}$.
Algorithm

1. For each correspondence $\mathbf{x_i} \leftrightarrow \mathbf{x'_i}$ compute $\mathbf{L_i}$. Usually only two first rows needed.
2. Assemble n 2×9 matrices $\mathbf{L_i}$ into a single $2n \times 9$ matrix $\mathbf{L}$.
3. Obtain SVD of $\mathbf{L}$ as $\mathbf{UDV^T}$ with $\mathbf{D}$ diagonal with positive diagonal entries, arranged in descending order down the diagonal, then $\mathbf{h}$ is last column of $\mathbf{V}$.
4. Determine $\mathbf{H}$ from $\mathbf{h}$.

i Transform the image coordinates according to the transformations $\tilde{\mathbf{x}}_\mathbf{i} = \mathbf{T}\mathbf{x_i}$ and $\tilde{\mathbf{x}}'_\mathbf{i} = \mathbf{T}'\mathbf{x'_i}$.

ii Find the transformation $\tilde{\mathbf{H}}$ from the correspondences $\tilde{\mathbf{x}}_\mathbf{i} \leftrightarrow \tilde{\mathbf{x}}'_\mathbf{i}$.

iii Set $\mathbf{H} = \mathbf{T}'\tilde{\mathbf{H}}\mathbf{T}$.

Hartley shows that data normalization gives dramatically better results and hence should be considered as an essential step in the algorithm [7]. One of the commonly used transformation is to translate the points so that their centroid is at the origin and the points are scaled such that the average distance from the origin is equal to $\sqrt{2}$.

3.1.2 DLT using line correspondences

Aggregate features such as lines and curves are easier to detect and match than point features. They can be used to obtain more robust correspondences. In the projective plane, points and lines are dual elements. Corresponding lines are related by $\mathbf{l}' = (\mathbf{H^{-1}})^\mathbf{T}\mathbf{l}$, or, switching left with right, as $\mathbf{l} = \mathbf{H}^\mathbf{T}\mathbf{l}'$. If we have a few corresponding lines, the above equations can be rearranged in matrix form to obtain:

$$\begin{bmatrix} .. & .. & .. & .. & .. & .. & .. & .. \\ t'_i & 0 & -t_i t'_i & u'_i & 0 & -t_i u'_i & 1 & 0 \\ 0 & t'_i & -u_i t'_i & 0 & u'_i & -u_i u'_i & 0 & 1 \\ .. & .. & .. & .. & .. & .. & .. & .. \end{bmatrix} \begin{bmatrix} H_{11} \\ H_{12} \\ H_{13} \\ H_{21} \\ H_{22} \\ H_{23} \\ H_{31} \\ H_{32} \end{bmatrix} = \mathbf{L} = \mathbf{0}. \tag{15}$$

This can be solved using a similar procedure as given above.

3.1.3 Using points and lines

It is clear that homography can be computed from a mixture of corresponding points and lines. Equations 14 and 15 can be combined and rearranged in the matrix form [25] to get:

$$\begin{bmatrix} .. & .. & .. & .. & .. & .. & .. & .. \\ x_i & y_i & 1 & 0 & 0 & 0 & -x_i x_i' & -y_i x_i' \\ 0 & 0 & 0 & x_i & y_i & 1 & -x_i y_i & -x_i y_i' \\ .. & .. & .. & .. & .. & .. & .. & .. \\ t_i' & 0 & -t_i t_i' & u_i' & 0 & -t_i u_i' & 1 & 0 \\ 0 & t_i' & -u_i t_i' & 0 & u_i' & -u_i u_i' & 0 & 1 \end{bmatrix} \begin{bmatrix} H_{11} \\ H_{12} \\ H_{13} \\ H_{21} \\ H_{22} \\ H_{23} \\ H_{31} \\ H_{32} \end{bmatrix} = \mathbf{L} = \mathbf{0}. \quad (16)$$

If there are more than four elements (points or lines), the above equations can be solved in a least-squares solution for $\mathbf{H}$ using the DLT procedure similar to the one given in Algorithm 1. The complete algorithm is given below.

Algorithm 2 Direct Linear Transformation of points and lines

Goal: Given 4 or more corresponding elements (points or lines) compute the homography between the two images.
Algorithm

1. For each correspondence $\mathbf{x_i} \leftrightarrow \mathbf{x_i'}$ or $\mathbf{l_i} \leftrightarrow \mathbf{l_i'}$ compute $\mathbf{L_i}$. Usually only two first rows needed.
2. Assemble n 2×9 matrices $\mathbf{L_i}$ into a single $2n \times 9$ matrix $\mathbf{L}$.
3. Obtain SVD of $\mathbf{L}$. Solution for $\mathbf{h}$ is last column of $\mathbf{V}$.
4. Determine $\mathbf{H}$ from $\mathbf{h}$.

Four matches in the form of points or lines are required to fix the 8 degrees of freedom in $\mathbf{H}$. However, there is a degenerate case that needs attention when using mixed correspondences. The homography can be computed from correspondences of 4 points, 3 points and 1 line, 3 lines and 1 point, or 4 lines, but not from the correspondences of two points and two lines [8]. Three lines and one point is geometrically equivalent to four points since three lines define a triangle and the vertices of the triangle uniquely defines three points. Similarly, the case of three points and a line is equivalent to four lines. However, the case of two points and two lines is equivalent to five lines with four concurrent or five points with four collinear. This configuration is degenerate and a one-parameter family of homographies map the two-point and two-line configuration to the corresponding configuration.

3.1.4 Robust estimation using RANSAC

So far it was assumed that the only source of error in the set of correspondences, $\mathbf{x_i} \leftrightarrow \mathbf{x_i'}$, is in the measurement of positions. In many practical situations this assumption is not valid because correspondences are computed automatically and are often mismatched. The mismatched points can be considered as outliers to a Gaussian distribution that explains the error in measurements. These outliers can severely disturb the estimated homography and should be identified. The goal then is to determine a set of inliers from the presented correspondences so that the homography can be estimated in an optimal manner from these inliers using the algorithm described in the previous section. This is robust estimation since the estimation is robust or tolerant to outliers, i.e., measurements following a different, and possibly unmodelled, error distribution.

The RANSAC algorithm [6] can be applied to the putative correspondences to estimate the homography and the (inlier) correspondences which are consistent with this estimate. The sample size is four, since four correspondences determine a homography. The number of samples is set adaptively as the proportion of outliers is determined from each consensus state.

Algorithm 3 RANSAC

Goal: Compute the homography between the two images given a set of candidate matches.
Algorithm

1. Select four points from the set of candidate matches, and compute homography.
2. Select all the pairs which agree with the homography. A pair ($\mathbf{x}$; $\mathbf{x}'$), is considered to agree with a homography H, if $d(\mathbf{H}\mathbf{x}; \mathbf{x}') < t$, for some threshold t and $d(.)$ is the Euclidean distance between two points.
3. Repeat steps 1 and 2 until a sufficient number of pairs are consistent with the computed homography.
4. Recompute the homography using all consistent correspondences.

There are some important issues in robust estimation using the above procedure [8]. The distance threshold t should be chosen, such that the point is an inlier with a probability α. This calculation requires known probability distribution for the distance of an inlier from the model. In practice, the distance threshold t is chosen empirically so that the probability α that the point is an inlier is high, such as, 0.95. Secondly, trying every possible sample may be prohibitively expensive. Instead a large number of samples is used so that at least one of the random samples of 4 points is free from outliers with a high probability, such as, 0.99. Another rule of thumb employed is to terminate the iterations if the size of the consensus set T is similar to the number of inliers believed to be in the data set. Given the assumed proportion of outliers, we can use $T = (1-t)n$ for n data points.

3.1.5 Statistical optimization

The least-squares method is widely used for its computational simplicity. However it can produce statistical bias because homogeneous coordinates are treated as if they were physical quantities without distinguishing measurement data from artificial constants. Since noise occurs in image coordinates subject to Euclidean geometry, it is preferable to work in the Euclidean framework. Kanatani et al. proposed one such method to compute the homography in the Euclidean framework [12]. The uncertainty of data points (x, y) and (x', y') is described by their covariance matrices $\mathbf{\Sigma}$ and $\mathbf{\Sigma}'$. It follows that the vectors $\mathbf{x}$ and $\mathbf{x}'$ have the following singular covariance matrices.

$$\mathbf{V}[\mathbf{x}_\alpha] = \frac{1}{f^2}\begin{bmatrix} \mathbf{\Sigma} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{V}[\mathbf{x}'_\alpha] = \frac{1}{f^2}\begin{bmatrix} \mathbf{\Sigma}' & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix}.$$

If the covariance matrices are known up to a scale, $\mathbf{V}[\mathbf{x}_\alpha] = \varepsilon^2 \mathbf{V_0}[\mathbf{x}_\alpha]$ and $\mathbf{V}[\mathbf{x}'_\alpha] = \varepsilon^2 \mathbf{V_0}[\mathbf{x}'_\alpha]$. The unknown magnitude ε can be considered to be the noise level. The normalized covariance matrices $\mathbf{V_0}[\mathbf{x}_\alpha]$ and $\mathbf{V_0}[\mathbf{x}'_\alpha]$ specify the relative dependence of noise occurrence on positions and orientations. If no prior knowledge is available for them, isotropy and homogeneity can be assumed. This translates to the default values $\mathbf{V_0}[\mathbf{x}_\alpha] = \mathbf{V_0}[\mathbf{x}'_\alpha] = diag(1, 1, 0)$. Let $\mathbf{H}'$ be an estimate of the homography, and $\mathbf{H}$ its true value. The uncertainty of the estimate $\mathbf{H}'$ is measured by its covariance tensor

$$V[\mathbf{H}'] = E\left[P((\mathbf{H}' - \mathbf{H}) \otimes (\mathbf{H}' - \mathbf{H}))P^T\right],$$

where $E[.]$ denotes expectation. The symbol $\otimes$ denotes tensor product. The $(ijkl)$ element of the tensor P is given by $P_{ijkl} = \delta_{ik}\delta_{jl} - H_{ij}H_{kl}$, where δ_{ij} is the Kronecker delta. Let us denote

$$\mathbf{W}_\alpha = \mathbf{x}'_\alpha \times \mathbf{H}\mathbf{V_0}[\mathbf{x}]\mathbf{H^T} \times \mathbf{x}'_\alpha + (\mathbf{H}\mathbf{x}_\alpha) \times \mathbf{V_0}[\mathbf{x}_\alpha] \times (\mathbf{H}\mathbf{x}_\alpha).$$

The theoretical accuracy bound can be attained in the first order by minimizing the squared Mahalanobis distance. Using Lagrange multipliers and first order approximation we get

$$\mathbf{J} = \sum \left(\mathbf{x}'_\alpha \times \mathbf{H}\mathbf{x}, \mathbf{W}_\alpha \left(\mathbf{x}'_\alpha \times \mathbf{H}\mathbf{x}_\alpha\right)\right).$$

The eigenmatrix $\mathbf{U}$ of the resulting covariance tensor $V[\mathbf{H}']$ for the maximum eigenvector λ indicates the orientation in the 9-dimensional space along which error is most likely to occur. The solution $\mathbf{H}$ is perturbed along that orientation in both directions as $\mathbf{H}^{(+)} = N[\mathbf{H}' + \sqrt{\lambda}\mathbf{U}]$ and $\mathbf{H}^{(-)} = N[\mathbf{H}' - \sqrt{\lambda}\mathbf{U}]$ The operator $N[.]$ denotes normalization to a unit norm.

The homography can be estimated by renormalization which is described as follows [12]:

i Let $c = 0$ and $\mathbf{W}_\alpha = \mathbf{I}, \alpha = 1, ..., N$.

ii Define the following tensor M

$$M = \frac{1}{N}\sum_{\alpha=1}^{N}\sum_{\mathbf{k},\mathbf{l}=1}^{3} \mathbf{W}_\alpha^{(\mathbf{kl})}(\mathbf{e}^{(\mathbf{k})} \times \mathbf{x}'_\alpha) \otimes \mathbf{x}_\alpha \otimes (\mathbf{e}^{(\mathbf{l})} \times \mathbf{x}'_\alpha) \otimes \mathbf{x}_\alpha. \tag{17}$$

iii Compute the following tensor $N = (N_{ijkl})$

$$N_{ijkl} = \frac{1}{N}\sum_{\alpha=1}^{N} \sum_{m,n,p,q=1}^{3} \epsilon_{imp}\epsilon_{knq}\mathbf{W}_\alpha^{\mathbf{mn}}(\mathbf{V_0}[\mathbf{x}_\alpha]_{jl}x'_{\alpha(p)}x'_{\alpha(q)} + \mathbf{V_0}[\mathbf{x}'_\alpha]_{jl}x_{\alpha(j)}x_{\alpha(l)}). \tag{18}$$

iv Compute the nine eigenvalues $\lambda_1 \geq ... \geq \lambda_9$ of tensor

$$\hat{M} = M - cN, \tag{19}$$

and the corresponding orthonormal system of eigenmatrices $\mathbf{H_1}$, ...,$\mathbf{H_9}$ of unit norm. If $\lambda_9 \approx 0$, stop. Else update c and $\mathbf{W}_\alpha$ in the following way and go back to step (ii) with

$$c \leftarrow c + \frac{\lambda_9}{(\mathbf{H_9}; N\mathbf{H_9})} \tag{20}$$

and

$$\mathbf{W}_\alpha \leftarrow \mathbf{x}'_\alpha \times \mathbf{H_9}\mathbf{V_0}[\mathbf{x}_\alpha]\mathbf{H_9^T} \times \mathbf{x}'_\alpha + (\mathbf{H_9}\mathbf{x}_\alpha) \times \mathbf{V_0}[\mathbf{x}'_\alpha] \times \mathbf{V_0}[\mathbf{x}'_\alpha] \times (\mathbf{H_9}\mathbf{x}_\alpha). \tag{21}$$

3.2 Points and lines with known epipolar geometry

There are two relations for points on a plane between the two views. First, a point in one view determines a line in the other view through the epipolar geometry (Equation 8). This line is the image in the second view of the ray through the first camera center and the image point. Second, a point in one view determines the point in the other through the homography (Equation 5). The point in the second view is the image of the intersection of the above ray with the given plane. This section presents results that tie together these two relations of 2-view geometry.

Epipolar geometry is defined only for cameras that have different principal points. Thus, these methods can be used only to recover the homography induced by planes and cannot be applied to recover the rotational homography.

Algorithm 4 Statistical optimization

Goal: Compute the homography between the two images given the correspondences between the noisy points $\mathbf{x_i}$ and $\mathbf{x'_i}$.
Algorithm

1. Initialize the weight matrix $\mathbf{W}_\alpha$ to an identity matrix.
2. Compute the tensors M and N using Equations 17 and 18.
3. Compute the nine eigenvalues of tensor $\hat{M}$ from Equation 19.
4. If the last eigenvalue is approximately zero then stop, else update c and $\mathbf{W}_\alpha$ using Equations 20 and 21 and repeat the above procedure.
5. The homography is the eigenmatrix $\mathbf{H_9}$ of tensor $\hat{M}$.

3.2.1 Three points

A plane in 3-space can be specified by three points. In turn these 3D elements can be specified by image correspondences. Thus, we do not need the coordinates of the plane to compute the homography. It can be computed directly from the corresponding image points that specify the plane. This is a natural mechanism to use in applications.

Given views of three (non-collinear) points $\mathbf{X_i}$ and the fundamental matrix $\mathbf{F}$, we can determine the constraint they place on the three-parameter family of homographies compatible with $\mathbf{F}$, as given by Equation 11. The problem is reduced to that of solving for $\mathbf{v}$ from the three point correspondences [8, 24]. Each correspondence $\mathbf{x_i} \leftrightarrow \mathbf{x'_i}$ generates a linear constraint on $\mathbf{v}$ as

$$\mathbf{x'_i} = \mathbf{H}\mathbf{x_i} = \mathbf{A}\mathbf{x_i} - \mathbf{e}(\mathbf{v^T}\mathbf{x_i}),\ \text{i = 1, 2, 3} \tag{22}$$

From Equation 22 the vectors $\mathbf{x'_i}$ and $\mathbf{A}\mathbf{x_i} - \mathbf{e'}(\mathbf{v^T}\mathbf{x_i})$ are parallel so their vector product is zero:

$$\mathbf{x'_i} \times (\mathbf{A}\mathbf{x_i} - \mathbf{e'}(\mathbf{v^T}\mathbf{x_i})) = (\mathbf{x'_i} \times \mathbf{A}\mathbf{x_i}) - (\mathbf{x'_i} \times \mathbf{e'})(\mathbf{v^T}\mathbf{x_i}) = \mathbf{0}.$$

Forming the scalar product with the vector $\mathbf{x'_i} \times \mathbf{e'}$ gives

$$\mathbf{x_i^T}\mathbf{v} = \frac{(\mathbf{x'_i} \times (\mathbf{A}\mathbf{x_i}))^\mathbf{T} (\mathbf{x'_i} \times \mathbf{e})}{(\mathbf{x'_i} \times \mathbf{e})^\mathbf{T} (\mathbf{x'_i} \times \mathbf{e})}. \tag{23}$$

which is linear in $\mathbf{v}$. Each correspondence generate an equation $\mathbf{x_i^T}\mathbf{v} = b$ and collecting together we have $\mathbf{L}\mathbf{v} = \mathbf{b}$.

Algorithm 5 Three points and epipolar geometry

Goal: Given $\mathbf{F}$ and three point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x'}_i$ which are the images of 3D points $\mathbf{X_i}$, determine the homography $\mathbf{x'} = \mathbf{H}\mathbf{x}$ induced by the plane of $\mathbf{X_i}$.
Algorithm

1. Compute the epipole $\mathbf{e'}$ using Equation 10.
2. Choose $\mathbf{A} = [\mathbf{e'}]_\times \mathbf{F}$ and solve linearly for $\mathbf{v}$ from $\mathbf{L}\mathbf{v} = \mathbf{b}$ as given in Equation 23.
3. Then $\mathbf{H} = \mathbf{A} - \mathbf{e'}\mathbf{v^T}$.

A solution cannot be obtained if $\mathbf{L^T} = [\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}]$ is not of full rank. Algebraically, $|\mathbf{L}| = \mathbf{0}$ if the three image points $\mathbf{x_i}$ are collinear. Geometrically, three collinear points arise from collinear world points, or coplanar world points where the planar world contains the first camera center. In either case, a full rank homography is not defined.

The algorithm requires the epipolar geometry to be known, thus more than just the four points are necessary for determining the homography matrix. At least 8 point correspondences determine the fundamental matrix if no other information is given. However, since usually at least 20-30 point correspondences are recommended for determining the homography in case of noisy data set, hence this constraint can be ignored in practical situations. A comprehensive review of the Fundamental Matrix Estimation techniques is given in [23].

3.2.2 A point and a line

A line correspondence reduces the three-parameter family of homographies compatible with $\mathbf{F}$ given in Equation 11 to a 1-parameter family [24, 26] given by

$$\mathbf{H}(\mu) = [\mathbf{l}']_\times \mathbf{F} + \mu \mathbf{e}' \mathbf{l}^T, \tag{24}$$

provided $\mathbf{l'^T e'} \neq 0$ where μ is a projective parameter. The point correspondence uniquely determines the plane and the corresponding homography. Thus, given the Fundamental matrix and a corresponding point $\mathbf{x} \leftrightarrow \mathbf{x}'$ and a line $\mathbf{l} \leftrightarrow \mathbf{l}'$, Luong [24] gives the method to compute homography as

$$\mathbf{H} = [\mathbf{l}']_\times \mathbf{F} + \frac{(\mathbf{x}' \times \mathbf{e}')^\mathbf{T} (\mathbf{x}' \times (\mathbf{F}\mathbf{x}) \times \mathbf{l}')}{\|\mathbf{x}' \times \mathbf{e}'\|^\mathbf{2} (\mathbf{l^T x})} \mathbf{e}' \mathbf{l^T}. \tag{25}$$

Algorithm 6 A point and a line and epipolar geometry

Goal: Given $\mathbf{F}$ and the a corresponding point $\mathbf{x} \leftrightarrow \mathbf{x}'$ and line $\mathbf{l} \leftrightarrow \mathbf{l}'$ compute the homography induced by the plane.

Algorithm

1. Compute the epipole $\mathbf{e}'$ from Equation 10
2. Compute the one parameter family for $\mathbf{H}$ using the line correspondence given by Equation 24.
3. Compute μ and $\mathbf{H}$ from the point correspondence $\mathbf{x} \leftrightarrow \mathbf{x}'$ using Equation 25.

3.3 Using conics

Point and line correspondences are not available in many situations or could be noisy. Higher order parametric curves make the problem easier as they can be recovered robustly from the image due to the large number of points defining them. Multiple views of a plane containing conics can help in recovering the homography between the views. Sugimoto dealt with homography calculation from conics using a minimum of 7 conic correspondences [31]. He used a 6×6 transformation relating conics and derived the homography in a linear fashion using it. Mudigonda et al. describe a method of computing the homography using only two conic correspondences [16]. Two given conics intersect at 4 points. If these points can be recovered and corresponded, the homography can

be calculated directly. If not, both images can be metric rectified by assuming that the given conics are images of circles. Note that this assumption is not valid when the conics intersect at 4 points because any two circles can intersect at a maximum of 2 real points. The method works even when the conics are not obtained by images of circles. Even though the rectification would be incorrect in this case, both the views are transformed incorrectly in a similar way and the final homography comes out to be correct.

Given a view with conics $\mathbf{C_1^1}$ and $\mathbf{C_1^2}$, obtain the homography $\mathbf{H_1}$ such that $\mathbf{C_1^{1\prime}} = \mathbf{H_1^{-T} C_1^1 H_1^{-1}}$ and $\mathbf{C_1^{2\prime}} = \mathbf{H_1^{-T} C_1^2 H_1^{-1}}$ where $\mathbf{C_1^{1\prime}}$ and $\mathbf{C_1^{2\prime}}$ are circles. Details of this step can be found in [16]. Similarly for the second view containing containing conics $\mathbf{C_2^1}$ and $\mathbf{C_2^2}$, the homography $\mathbf{H_2}$ is obtained such that $\mathbf{C_2^{1\prime}} = \mathbf{H_2^{-T} C_2^1 H_2^{-1}}$ and $\mathbf{C_2^{2\prime}} = \mathbf{H_2^{-T} C_2^2 H_2^{-1}}$ where $\mathbf{C_2^{1\prime}}$ and $\mathbf{C_2^{2\prime}}$ are circles. If the views are of the same scene, there exists a similarity transform $\mathbf{H_s}$ such that

$$\mathbf{C_1^{1\prime}} = \mathbf{H_s^{-T} C_2^{1\prime} H_s^{-1}} \text{ and } \mathbf{C_1^{2\prime}} = \mathbf{H_s^{-T} C_2^{2\prime} H_s^{-1}}.$$

We can solve for $\mathbf{H_s}$ using two point correspondences, such as the centers of the circles in the rectified view as similarity transform preserves the centers. The homography between the two views is the product $\mathbf{H_1 H_s H_2^{-1}}$. We can obtain a unique $\mathbf{H_s}$ if the radii of the two circles are different. If the radii are the same, we cannot determine which center in one view corresponds to which center in the other. This results in the rotation and translation being unknown. If the circles are concentric, the scale and translation can be determined from the radii and the center, but the rotation ambiguity will remain. This approach involves simple univariate polynomial equations of degree 4 which always have a solution.

Algorithm 7 From a pair of conics

Goal: Given 2 corresponding conics compute the homography between the two images.
Algorithm

1. Obtain the equations of the conics in both views.
2. Rectify both views assuming that the conics are images of circles. The results are correct even if this is not so. Let $\mathbf{H_1}$ and $\mathbf{H_2}$ be the rectifying homographies.
3. Calculate the similarity transform $\mathbf{H_s}$ between the two rectified views using two point correspondences obtained by finding the centers of the two circles.
4. The homography between the two views is obtained as $\mathbf{H_1 H_s H_2^{-1}}$.

Conics are higher order features and the homography is recovered using the algebraic constraints on matching conics. Hence, this method can be used to recover rotational homogrphy as well as planar homography.

3.4 Using curves

Since scenes rich with man-made objects contain curve-like features, the next natural step has been to consider higher order curves. A single curve match between two images is sufficient for solving for the associated homography matrix.

The result presented in Section 3.4.1 can be used to recover homography due to planes and rotation. The result presented in Section 3.4.2 depends on the epipolar geometry and the epipoles and cannot recover rotational homographies.

3.4.1 Algebraic curves

The general issue of multiview geometry of algebraic curves for recovering homography matrix is addressed by Kaminski and Shashua in [10]. Let $\mathbf{C}$ (respectively $\mathbf{C}'$) be the tensor form of the first (second) image curve. Let A be the tensor form of the homography matrix. Then

$$\exists \lambda \neq 0, \text{ such that: } \mathbf{C}'_{i_1,\ldots,i_n} A^{i_1}_{j_1} \ldots A^{i_n}_{j_n} = \lambda \mathbf{C}_{j_1,\ldots,j_n}.$$

Since a planar algebraic curve of order n is represented by a polynomial containing $\frac{1}{2}n(n+3)+1$ terms, each one provides a total of $\frac{1}{2}n(n+3)$ equations (after elimination of λ) on the entries of the homography matrix. Let S denote the system of equations. Two curves of order $n \geq 3$ are sufficient to recover the epipolar geometry. However a simpler algorithm exists for non-oversingular curves. A curve of order n whose only singular points are either nodes or cusps satisfies the Plucker's formula $3n(n-2) = i + 6\delta + 8\kappa$, where i is the number of inflexion points, δ is the number of nodes and κ is the number of cusps. A curve is said to be *non-oversingular* when its only singularities are nodes and cusps and when $i + s \geq 4$, where s is the number of all singular points. To compute the homography first compute the intersection of these curves with the Hessian curve. The Hessian curve is given by the determinant equation

$$\left|\frac{\partial^2 f}{\partial x_i \partial x_j}\right| = 0, \tag{26}$$

where $f(x_1, x_2, x_3) = 0$ gives the equation of the curve. The intersection points so computed are in fact the singular and inflexion points.

At first sight there are $i! \times s!$ possible correspondences between the sets of inflexion and singular points in the two images. But it is possible to further reduce the combinatorics by separating the points into two categories. The points are normalized such that the last coordinates is 1 or 0. Then real points from complex points are separated. Each category of the first image must be matched with the same category in the second image. Then the right solution can be selected as it should be the one that makes the system S the closest to zero or the one that minimizes the Hausdorff distance [9] between the set of points from the second image curve and the reprojection of the set of points from the first image curve into the second image.

Algorithm 8 Hessian Curve based

Goal: To compute the homography between the two views given a pair of cubic or higher order curves between them.

Algorithm

1. Compute the Hessian curves in both images using Equation 26.
2. Compute the intersection of the curve with its Hessian in both images. The output is the set of inflexion and singular points.
3. Discriminate between inflexion and singular points by the additional constraint for each singular point $\nabla f(\mathbf{a}) = 0$.
4. Separate the real points from the complex points.
5. Find the solution to $\mathbf{H}$ that makes S the closest to zero or minimizes the Hausdorff distance between the set of points.

3.4.2 Non-algebraic curves

Schmid and Zisserman show how Euclidean curvature is mapped by a homography and present a method to estimate the homography from it [27, 28]. The Euclidean curvatures κ and κ' at matching points $\mathbf{x}$ and $\mathbf{x}'$ are related in this case by

$$\kappa' = \kappa \left(\frac{ds}{ds'}\right)^3 \left(\frac{1}{\mathbf{h^3}.\mathbf{x}}\right)^3 |\mathbf{H}|.$$

where $|\mathbf{H}|$ is the determinant of $\mathbf{H}$ and $\frac{ds}{ds'}$ is the ratio of the arc lengths. If the tangent lines at $\mathbf{x}$, $\mathbf{x}'$ are $\mathbf{l}$, $\mathbf{l}'$ respectively. Given the epipolar geometry, there is a one line parameter family of homographies given by Equation 24. The notation $\dot{\mathbf{x}}$ indicates the derivative of the homogeneous vectors $\mathbf{x}$ by a curve parameter. Note that since $\mathbf{l} = \mathbf{x} \times \dot{\mathbf{x}}$ and $A = [l']_\times F$ it follows that $\mathbf{h_i x} = \mathbf{a_i x}$. After some algebra, it can be seen that

$$\mu\{l_1|\mathbf{e}'\mathbf{a_2}\mathbf{a_3}| + l_2|\mathbf{e}'\mathbf{a_1}\mathbf{a_3}| + l_3|\mathbf{e}'\mathbf{a_1}\mathbf{a_2}|\} = \frac{\kappa'}{\kappa}\left(\frac{ds'}{ds}\right)^3 (\mathbf{a_3}.\mathbf{x})^3, \tag{27}$$

where $\mathbf{a_i}$ is the ith column of $\mathbf{A}$, and $|\mathbf{xyz}|$ is the determinant of the matrix with columns $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$. Given two

---

Algorithm 9 Curvature-based

Goal: To compute the homography between the two views given the Fundamental Matrix and the perspective images of the plane curve and one corresponding image points

Algorithm

1. Compute curvature and tangents at $\mathbf{x}$ and $\mathbf{x}'$.
2. Compute the one-parameter family of the homography by the tangents of the curve at $\mathbf{x}$ and $\mathbf{x}'$ using Equation 24.
3. Choose $\mathbf{A} = [\mathbf{e}']_\times \mathbf{F}$.
4. The parameter μ can be determined using the curvature at points $\mathbf{x}$ and $\mathbf{x}'$ and $\mathbf{A}$ using Equation 27.
5. Compute $\mathbf{H}$ from Equation 24.

---

perspective images of a plane curve and the fundamental matrix between the views, the plane of the curve (and consequently the homography induced by this plane) is defined uniquely from the corresponding tangent lines and curvature at one point. There are two types of degenerate points at which the plane/homography cannot be uniquely determined by this method. The first is at epipolar tangents, where the tangent lines contain the epipole (i.e. if $\mathbf{l}.\mathbf{e} = \mathbf{l}'.\mathbf{e}' = \mathbf{0}$), since in this case the 3D line which projects to $\mathbf{l}$ and $\mathbf{l}'$ cannot be determined as it lies in an epipolar plane. The second is at zero's of curvature (inflections) since the curvature cannot be used to determine μ.

3.5 Using discrete contours

Discrete contours are non-algebraic contours or shape-boundaries which are represented by a sequence of points $\mathbf{x}[i]$ in clockwise or anti-clockwise order. Discrete non-parametric contours can be found more commonly in everyday images than parametric or algebraic curves like conics. If the image-to-image homography is affine, the transformation matrix has $H_{31} = H_{32} = 0$ and $H_{33} = 1$. Thus, the transformation can be given as

$$\mathbf{H} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{28}$$

In terms of inhomogeneous coordinates it can be expressed as

$$\mathbf{x}'[i] = \mathbf{A}\mathbf{x}[i] + \mathbf{b},$$

where $\mathbf{x}[i]$ and $\mathbf{x}'[i]$ are the inhomogeneous representations of the ith point on the contour. If correspondence information is not available,

$$\mathbf{x}'[i] = \mathbf{A}\mathbf{x}[i + \lambda] + \mathbf{b}.$$

The Fourier domain representation can be given for non DC terms by

$$\mathbf{X}'[k] = \mathbf{A}\mathbf{X}[k]e^{(\frac{j2\pi\lambda k}{N})}, 0 < k < N.$$

We can define measures κ and κ' on the sequences $\mathbf{X}$ and $\mathbf{X}'$ as

$$\kappa[k] = \mathbf{X}[k]^{*T} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \mathbf{X}[k] \text{ and } \kappa'[k] = \mathbf{X}'[k]^{*T} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \mathbf{X}'[k].$$

It can be shown that

$$\kappa'[k] = |\mathbf{A}|\ \kappa[k], \tag{29}$$

if sequences $\mathbf{x}$ and $\mathbf{x}'$ are affine-related [15, 18].

Similarly,

$$\kappa'_p[k] = \mathbf{X}'[k]^{*T} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \mathbf{X}'[p] \tag{30}$$

$$= |\mathbf{A}|\kappa[k]e^{-j\omega\lambda(p-k)} \tag{31}$$

The shift λ can be computed from the inverse Fourier transform of this quotient series. The algorithm to compute the affine homography between two views using a pair of discrete contours is given in below.

Algorithm 10 Affine Homography from Discrete non-parametric contours

---

Goal: Given a pair of matching discrete non-parametric contours between two views find the affine homography between them.

Algorithm

1. Compute the DFT of the contour points $\mathbf{x}'$ and $\mathbf{x}$.
2. Check if the contours match by computing the κ values and verifying if Equation 29 is satisfied.
3. Calculate λ.
4. Compute $\mathbf{A}$ matrix using a linear least squares solution if they match.
5. Determine the scale for $\mathbf{A}$ using the average distance to points from the centroid.
6. Compute $\mathbf{b}$ from the difference in centroids.
7. Determine $\mathbf{H}$ from Equation 28.

The projective homography also can be estimated from a single pair of matching discrete contours [19]. Define measure $\lambda(z)$ as the ratio of the cross-ratio of areas. Let $\mathbf{x}$ be the given point in the first view. Let $\mathbf{y}$ be the point

adjacent to $\mathbf{x}$ on the contour and $\mathbf{t_1}, \mathbf{t_2}, \mathbf{t_3}$ be any three points. Then

$$\lambda(z) = \frac{cr(\mathbf{x}, \mathbf{t_1}, \mathbf{t_2}, \mathbf{t_3}, \mathbf{y})}{cr(\mathbf{x}, \mathbf{t_1}, \mathbf{t_2}, \mathbf{t_3}, \mathbf{z})}.$$

The measure $|1 - \lambda(\mathbf{z})|$ provides an estimate of the deviation of $\mathbf{z}$ from the line joining $\mathbf{x}$ and $\mathbf{y}$. The invariance of $\lambda(\mathbf{z})$ is independent of $\mathbf{t_1}, \mathbf{t_2}, \mathbf{t_3}$ because its value will be 1 if $\lambda(\mathbf{z})$ lies on the line joining $\mathbf{x}$ and $\mathbf{y}$ regardless of the position of $\mathbf{t_1}, \mathbf{t_2}$ or $\mathbf{t_3}$. These invariant lines can be used to obtain large number of correspondences on the contour very robustly.

Define $\mu(z) = \text{dist}_z^{xt_3}/\text{dist}_z^{xt_2}$ where dist_p^{qr} is the distance from point p to the line connecting points q and r. $\mu(z)$ is invariant under projective transformation up to scale, i.e., the ratio of the values of $\mu(z)$ in two views is a constant for all points $\mathbf{z}$. Using the two invariant lines obtained, $\mu(z)$ can be calculated for all points on the contour. Given two views of the same object, two sequences of real numbers which are related by a common scale factor are obtained. A large number of correspondences on the contour can also be obtained from the common scale. However, the scale factor can be obtained more robustly from a Fourier domain representation of the sequence μ. Once the correspondences on the contours are available, the homography relating the two views can be computed by the DLT algorithm (Algorithm 1). The only case where the approach fails is if the contour is a triangle where two of the sides are selected as invariant lines. Therefore, all the dense correspondences will lie on the third side, i.e., all correspondences will be collinear and a unique solution will not be obtained. This case can be detected and a triangle specific homography procedure can be applied. The procedure listed above assumes that one pixel correspondence is known and uses them to define the the starting points. This makes it difficult to include this algorithm in automatic applications. We can, therefore, use the affine homography that can be obtained without known correspondence to bootstrap the process. The affine homography obtained as above can be used to obtain a pair of starting points needed to compute the projective homography.

Algorithm 11 Projective homography from discrete non-parametric contours

---

Goal: Given a pair of matching discrete non-parametric contours between two views find the homography between them.

Algorithm

1. Match and compute the affine homography between the given contours using Algorithm 10.
2. Take a pair of matching points in views 0 and 1 using the affine homography.
3. Find the invariant lines in both views by following the contour in clockwise and anti-clockwise directions.
4. Compute dense correspondences on the contour using the $\mu(\mathbf{z})$ measures.
5. Compute the common scale factor in the Fourier domain.
6. Compute the projective homography using a DLT algorithm.

The above method can recover homography induced by the plane as well as the rotational homography as it only uses the properties of matching sequences of points on the discrete contours.

3.6 Using planar texture

Identification of corresponding primitives in two images limits the applicability of above approaches. The texture based approach makes use of image intensities for computing the image-to-image transformation. Mudigonda et al. describe an algorithm for estimating the homography from Fourier Domain representation of images of planes [13, 17, 21]. The images are transformed to the Fourier domain and then represented in a coordinate system in which the affine transformation is reduced to an anisotropic scaling. The anisotropic scale factors can be computed using cross correlation methods, and working backwards from this, the entire transformation is computed. The technique does not need any correspondence information across views. Kruger and Calway [14] described a similar multiresolution approach to register images under affine transformations.

If $\mathbf{H}$ is the affine transformation between the spatial representations of the two images, then the upper 2×2 minor of $\mathbf{H}^{-T}$ relates their 2D Fourier magnitude spectra [17]. Two corresponding lines passing through the origin in the Fourier magnitude spectra of the two views can be identified and used to define a new coordinate space, called the α-β space. Let L_i^j be the ith line in view j. To determine the coordinates in the α-β space of a point (u^j, v^j) in the Fourier magnitude spectra of in view j, draw lines l_1^j and l_2^j parallel to L_1^j and L_2^j respectively, passing through (u^j, v^j). The distance of (u^j, v^j) from the L_1^j along l_2^j be α^j and the distance from L_2^j along l_1^j be β^j. Represent each point (u^j, v^j) in the Fourier magnitude spectra space by (α^j, β^j) in α-β space.

It can be shown that two projectively views differ only by anisotropic scale factors in this space [17]. Logarithm of the representation in the α-β space transforms the scaling into a translation, which can be recovered by looking for an impulse in the inverse Fourier transform of the cross power spectrum. The location of the impulse gives the translation, the inverse transformation of which would yield the non-linear scale factors of ψ_1 and ψ_2.

Corresponding points in the α-β space can be obtained after on-linear scale factors are identified. From them, the corresponding points in the Fourier domain and the original image space can be obtained. From these corresponding points, the 2×2 minor of $\mathbf{H^{-1}}$ in a linear least squares manner can be computed. This transformation is then applied to the second image to get the transformed image $g_t(.)$. The unknown translation components can be determined from the location of the impulse in the inverse Fourier transform of the cross power spectrum of the original image $f(.)$ and $g_t(.)$. Lucchese uses the computed affine approximation of the transformation as an initial estimate in a non-linear optimization procedure to compute the projective components of the transformation [22]. The technique presented in [17] produces a far more robust estimate of the affine transformation and provides good initial estimates for an identical non-linear optimization procedure to compute the complete projective transformation.

The relationship between the textures can be used to recover homography induced by the plane as well as the rotational homography as it depends only on the matching of points and their effect in the frequency domain.

4 Experimental Comparison

In this section, we experimentally compare the different homography estimation algorithms on a synthetic data set. The synthetic data set allows the study of impact of noise, presence of multiple features, etc. All algorithms were implemented by us as per material presented in the respective papers. No attempts were made to carefully optimize any algorithm and better results could be obtained by careful tuning of each algorithm. However, the results presented here do bring out the major overall characteristics of each algorithm relative to the others. Some aspects of the experimental comparison are discussed below.

Algorithm 12 Homography Computation using textural information

Goal: Compute the homography between the two transformed views of the same planar texture.
Algorithm

1. Compute the DFT of both the images.
2. Transform the images to the $\alpha\beta$ space.
3. Recover the translation by looking for an impulse in the inverse Fourier Transform of the two cross power spectrum of the $\log \alpha\beta$ space representation of the two images which gives us the non-linear scale factors ψ_1 and ψ_2.
4. In the Fourier domain recover the upper 2×2 minor of $\mathbf{H^{-1}}$ in a linear least squares manner.
5. Recover the translation component from the location of the impulse in the inverse Fourier transform of the cross power spectrum of the original image $f(.)$ and $g_t(.)$.
6. Use Levenberg-Marquardt algorithm to compute the projective components of $\mathbf{H}$.

4.1 The data set

The data set consists of synthetic images of resolution 400×400 with the required ground truth in the form of the homography between planes in them. The images contained features such as points, lines, conics, cubic curves, and contours. For each, different noise levels could be added. We computed the homography using 30 elements in case of points and lines in the default situation. Studies were also conducted on the effect of the number of features on the recovered homography. The curve and contour based algorithms used only one pair for homography estimation as they already contained an aggregation of hundreds of points. The computed homography was then used to compute the squared error for 20 points of a test set.

The first view had the camera kept at a distance of 4 units along the Z-direction and perpendicular to the plane on which the points lie. The second view had the camera translated by 2.5 units in the negative X (horizontal) direction and rotated about the Y (vertical) axis by an angle of 20° with respect to the first camera. The random points were generated on the world plane and projected to subpixel accuracy using the camera matrices to these views. For noise studies, the points in the second view were displaced by adding the requisite Gaussian noise. The average distance between points and lines, wherever applicable, was kept around 110 pixels. The fundamental matrix $\mathbf{F}$ was computed directly using the two camera matrices and made available to algorithms that used the epipolar geometry. Thus, the results of homography estimation can be studied independent of the complications in the estimation of $\mathbf{F}$ from images. For homography estimation, the points of each data set were given to the program. Higher order features like lines, conics, curves, were estimated using a linear least squares method from the images.

4.2 The error function

For evaluating the algorithm, we use the error in the second image after transferring points from the first image using the computed homography matrix. This is the Euclidean image distance in the second image between the measured point $\mathbf{x}'$ and the point $\mathbf{Hx}$ to which the corresponding point $\mathbf{x}$ is mapped. If $d(\mathbf{x}, \mathbf{y})$ is the Euclidean distance between the inhomogeneous points represented by $\mathbf{x}$ and $\mathbf{y}$, the transfer error for the set of correspondences

Noise σ	DLT (Alg 1)	Lines & pts (Alg 2)	RANSAC (Alg 3)	Statistical (Alg 4)	Points & $\mathbf{F}$ (Alg 5)	Lines & $\mathbf{F}$ (Alg 6)
0	0.0000	0.0032	0.0000	0.0000	0.0034	0.0028
0.5	0.7031	0.7145	0.7656	0.68	2.2428	3.3352
1	1.4038	1.4140	1.386	1.384	5.7860	9.5295
2	3.6706	3.6374	3.6181	2.712	11.9607	16.2887
Quantization	0.5399	0.5454	0.5694	0.52	11.0226	6.3104

Noise σ	Conics (Alg 7)	Curvature (Alg 9)	Hessian (Alg 8)	Contour (Alg 11)	Texture (Alg 12)
0	0.0001	0.0140	0.0034	0.00	0.0004
0.5	0.80	28.8	51.3	0.64	0.03
1	1.78	31.5	56.3	1.24	0.7
2	2.86	38.5	75.7	2.42	43
Quantization	0.82	36.2	37	0.52	-

Table 1: Comparison of different algorithms for varying noise levels and quantization

is

$$E = \sum_i d(\mathbf{x}'_i, \mathbf{H}\mathbf{x}_i)^2.$$

Other error measures such as the symmetric transfer error and reprojection error can also be used for evaluating the algorithm. Our initial experiments with them give very similar results.

4.3 Homography estimation

Table 1 compares the performance of the above algorithms in the presence of Gaussian noise. Results for four levels of noise are shown in the table. The results are also plotted as a graph in Figure 1. We also show the impact of quantization, i.e., rounding the coordinates of the points to the nearest integer value. The results presented are the average figures for 5 sets of data points generated randomly and independently.

The performance of all the algorithms degrades as the noise level increases. The DLT algorithm using points, lines and their combinations, the RANSAC algorithm and the statistical optimization algorithms perform well and degrade gracefully with noise. Algorithm 4 by Kanatani gives the best results among all the algorithms. The conic and contour based algorithms perform well under increasing noise and quantization. The reasons could be that they capture aggregate features which hold up under noise. The curve-based algorithms show poor results as the noise level increases because of the error in the estimation of their parameters. The curve-based and Hessian-based algorithms depend critically on fitting higher order parameters from images. Hence they are very unstable under noisy conditions. Texture based algorithm gives the good results under low noise but degrades sharply as noise increases. The noise is applied on the gray level values of all pixels – and not pixel locations – for the methods that use texture.

Quantization adversely affects all algorithms, but the curve based algorithms degrade sharply due to their dependence on the parameters that are to be estimated from the images.

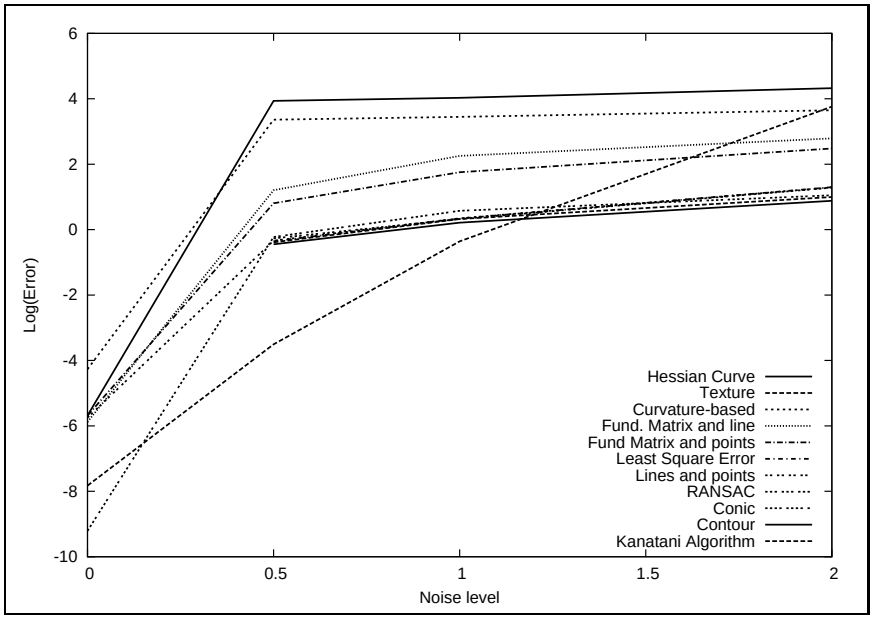

Figure 1: Plot of the log error against the σ of the noise for all algorithms

4.4 Varying features and their properties

Number of points	DLT (Alg 1)	Lines & pts (Alg 2)	RANSAC (Alg 3)	Statistical (Alg 4)
30	0.70	0.71	0.76	0.68
50	0.67	0.70	0.67	0.532
70	0.64	0.69	0.66	0.530

Table 2: Performance of the point/line algorithms with varying number of features

We compared the performances of the algorithms using lines and points when their number is increased. The level of noise was kept constant with zero mean and a variance of 0.5. Table 2 shows the results. The performance of the algorithms seem to improves, though slowly, with the number of features. This is expected as the additional features are from the same statistical distribution and carry the same amount of information. On iterative algorithms, the performance as the number of iterations is increased is shown in Table 3. The effect on the peformance is marginal but for the curve based algorithms.

Table 4 and Figure 3 show the performance of these algorithms as the average distance between features is increased, i.e., when they spread out wider. A Gaussian noise with a variance of 0.5 was added to coordinates and the number of features used was 30. The performance seems to improve slightly as the points become more widespread.

4.5 Changing view parameters

We studied these algorithms with varying distances and angles between the cameras. The distortion increases as the distance and angle are increased. Table 5 presents the results for the algorithms for 3 transformations for a

Number of Iterations	Points & **F** (Alg 5)	Lines & **F** (Alg 6)	Conics (Alg 7)	Curvature (Alg 9)	Hessian (Alg 8)	Contour (Alg 11)
1	0.85	2.73	1.02	17.7	54.2	0.7
3	0.80	2.7	1.0	17.7	43	0.62
5	0.80	2.68	0.965	14.4	43	0.62

Table 3: Performance of iterative algorithms with varying number of iterations

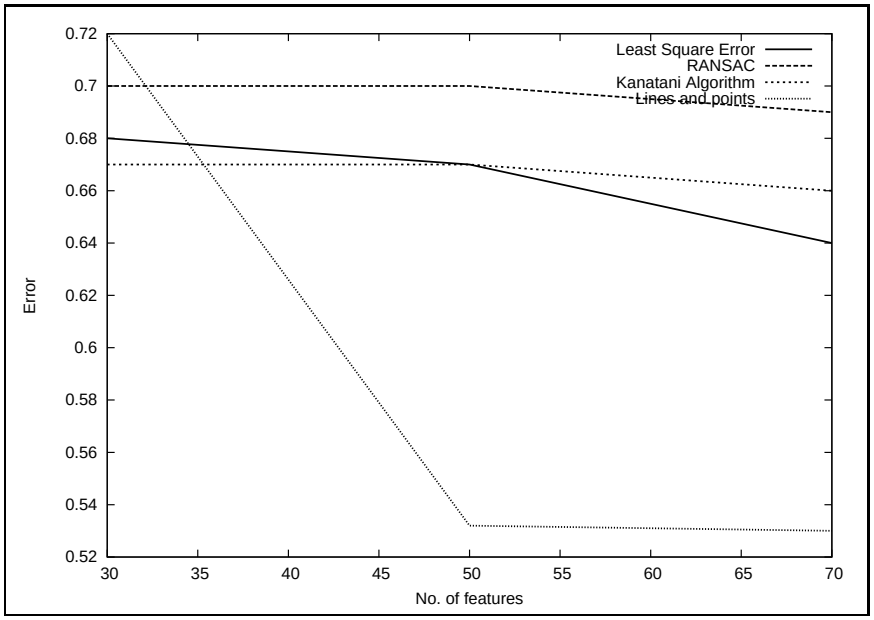

Figure 2: Plot of the error against the number of features

noise level of 0.5. The performance degrades slowly as the views become more distant. The degradation is more for the curve based algorithms due to the difficulty in estimating their parameters.

5 Conclusions

We presented a survey of the different techniques to estimate the planar homography between two images in this paper. The literature on homography estimation is quite mature and many techniques are available for the step, each with its own set of assumptions. We also presented an experimental comparison of the methods discussed

Mean distance between features	DLT (Alg 2)	RANSAC (Alg 3)	Statistical (Alg 4)
100	0.78	0.80	0.75
110	0.71	0.76	0.68
120	0.67	0.65	0.63

Table 4: Performance of the algorithms as the features are spread wider

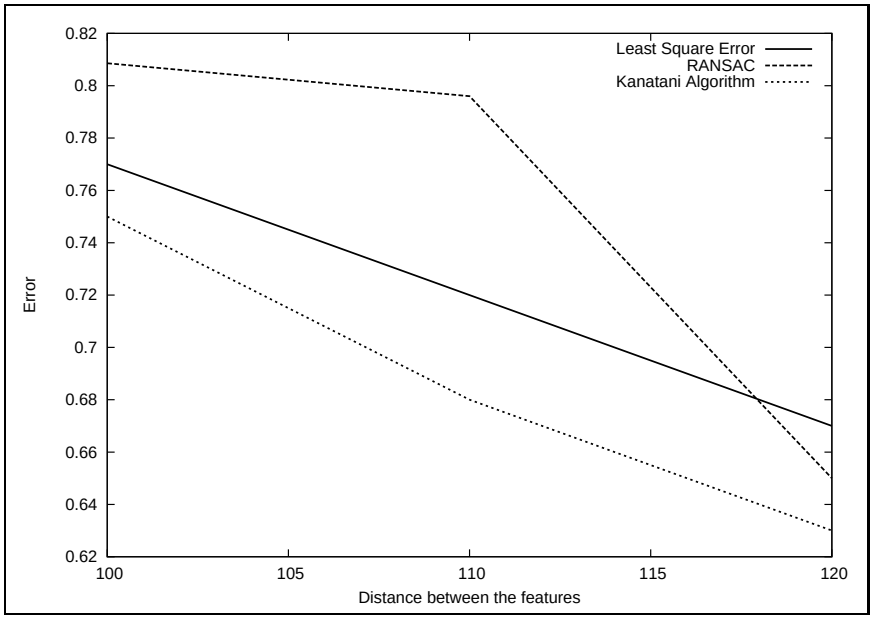

Figure 3: Plot of the error against the mean distance between features

Angle, Translation	DLT (Alg 1)	Lines & pts (Alg 2)	RANSAC (Alg 3)	Statistical (Alg 4)	Points & **F** (Alg 5)	Lines & **F** (Alg 6)
5, 1.0	0.71	0.7224	0.72	0.65	2.1	3.6255
20, 2.5	0.77	0.7598	0.76	0.68	2.24	4.6710
35, 4.0	0.77	1.6624	0.76	0.74	2.84	4.686

Angle, Translation	Conics (Alg 7)	Curvature (Alg 9)	Hessian (Alg 8)	Contour (Alg 11)	Texture (Alg 12)
5, 1.0	0.58	45.9	48.2	0.58	0.02
20, 2.5	1.08	50.2	56	0.64	0.04
35, 4.0	1.2	56	94	0.68	0.9

Table 5: Performance of the algorithms as transformation between the views is changed

on plausible images where the parameters could be varied. The results show that the performance is good when a large number of features are used, as with most optimization techniques. Techniques that use collections of points such as conics and discrete contours as features perform well. The algorithms that require the epipolar geometry as a precondtion and those using high order curves tend to perform poorly. This may be due to compounding of the uncertainty in the estimation of these primary structures before computing the homography.

Acknowledgements: We thank Sujit Kuthirummal and Pawan Mudigonda for their comments and help with the organization of the paper and the code used for some of the experiments.

References

[1] E. H. Adelson and J. R. Bergen. The plenoptic function and the elements of early vision. In *Computational Models of Visual Processing*. 1991.

[2] M. Brown, A. Majumder, and R. Yang. Camera-based calibration techniques for seamless multi-projector displays. *IEEE Transactions on Visualization and Computer Graphics*, 11(2):193–206, 2005.

[3] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. *International Journal of Computer Vision*, 40(2):123–148, 2000.

[4] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, 1993.

[5] O. Faugeras and Q. T. Luong. *The Geometry of Multiple Images*. MIT Press, 2001.

[6] 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 the ACM*, 24(6):381–395, 1981.

[7] R. Hartley. In defence of the 8-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593, 1997.

[8] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[9] D. Huttenlocher, D. Klanderman, and A. Rucklige. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863, 1993.

[10] J. Y. Kaminski and A. Shashua. Multiple view geometry of general algebraic curves. *Int. J. Comput. Vision*, 56(3):195–219, 2004.

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

[12] K. Kanatani, N. Ohta, and Y. Kanazawa. Optimal homography computation with a reliability measure. *Proceedings of the IAPR workshop on Machine Vision Applications*, 1998.

[13] S. Kruger and A. Calway. Image registration using multiresolution frequency domain correlation. In *Proc. of British Machine Vision Conference (BMVC)*, 1998.

[14] S. Kruger and A. Calway. Image registration using multiresolution frequency domain correlation. In *Proc. British Machine Vision Conference*, pages 316–325, 1998.

[15] M. P. Kumar, S. Goyal, S. Kuthirummal, C. V. Jawahar, and P. J. Narayanan. Discrete contours in multiple views: Approximation and recognition. *Image and Vision Computing*, 2004.

[16] M. P. Kumar, C. V. Jawahar, and P. J. Narayanan. Geometric structure computation from conics. In *Proc. Indian Conference on Computer Vision, Graphics, and Image Processing*, 2004.

[17] M. P. Kumar, S. Kuthirummal, C. V. Jawahar, and P. J. Narayanan. Planar homography from fourier domain representation. In *Proc. International Conference on Signal Processing and Communications (SPCOM)*, 2004.

[18] S. Kuthirummal, C. V. Jawahar, and P. J. Narayanan. Planar shape recognition across multiple views. In *Proc. International Conference on Pattern Recognition*, volume 1, pages 456–459, 2002.

[19] S. Kuthirummal, C. V. Jawahar, and P. J. Narayanan. Planar homography from discrete contours. Technical Report 001, International Institute of Information Technology, Hyderabad, India, 2003.

[20] D. Liebowitz and A. Zisserman. Metric rectification for perspective images of planes. In *Proc. Computer Vision and Pattern Recognition*, 1998.

[21] L. Lucchese. A frequency domain technique based on energy radial projections for robust estimation of global 2d affine transformations. *Computer Vision and Image Understanding*, 81:71–116, 2001.

[22] L. Lucchese. A hybrid frequency-space domain algorithm for estimating projective transformations of color images. In *Proc. International Conference on Image Processing*, volume 2, pages 913–916, 2001.

[23] Q. Luong and O. Faugeras. The fundamental matrix: theory,algorithms and stability analysis. In *Proc. European Conference on Computer vision*, pages 577–588, 1994.

[24] Q. Luong and T. Vieville. Canonical representation for the geometries of multiple views. *Computer Vision and Image Understanding*, 64(2):193–229, 1996.

[25] V. Murino, U. Castellani, A. Etrari, and A. Fusiello. Registration of very time-distant aerial images. In *Proc. International Conference on Image Processing*, pages 989–992, 2002.

[26] C. Schmid and A. Zisserman. Automatic line matching across views. In *Proc. Computer Vision and Pattern Recognition*, pages 666–671, 1997.

[27] C. Schmid and A. Zisserman. The geometry and matching of curves in multiple views. In *Proc. European Conference on Computer Vision*, 1998.

[28] C. Schmid and A. Zisserman. The geometry and matching of lines and curves over multiple views. *International Journal of Computer Vision*, 2000.

[29] J. G. Semple and G. T. Kneebone. *Algebraic Projective Geometry*. Oxford University Press, 1952.

[30] H.-Y. Shum and R. Szelishi. Construction of panoramic image mosaics with global and local alignment. *International Journal of Computer Vision*, 36(2):101–130, 2000.

[31] A. Sugimoto. A linear algorithm for computing the homography from conics in correspondence. *Journal of Mathematical Imaging and Vision*, 13:115 – 130, 2000.

[32] I. E. Sutherland. Sketchpad : A man-machine graphical communications system. Technical Report 296, MIT Lincoln Laboratories, 1963.

[33] R. Szelishi and H.-Y. Shum. Creating full view panoramic image mosaics and environment maps. In *Proc. Computer Graphics (SIGGRAPH 97)*, pages 251–258, 1997.

[34] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.