Master of Science by Research Thesis

Motion in Multiple Views

by

Sujit Kuthirummal

200207003

International Institute of Information Technology

Gachibowli, A.P., India. 500 019.

sujit@gdit.iiit.net

Advisors

Dr. C. V. Jawahar (jawahar@iiit.net)

Dr. P. J. Narayanan (pjn@iiit.net)

July 19, 2003.

Contents

List of Figures

List of Tables

Abstract

Mathematically, an image is the projection of the 3D world onto the 2D plane of the camera. This projection results in the loss of information present in the third dimension, popularly referred to as the depth or the z dimension. It is easy to see that a plurality of projections can compensate for this loss and this has led to the study of the geometry that underlies multiple views of the same scene.

Multiview analysis of scenes is an active area in Computer Vision today. The structure of points and lines as seen in two views attracted the attention of computer vision researchers like Longuet-Higgins and Oliver Faugeras in the eighties and early nineties. Similar studies on the underlying constraints in three or more views followed. The mathematical structure underlying multiple views has been analysed with respect to projective, affine, and euclidean frameworks of the world with amazing results. These multiview relations have been used for visual recognition of 3D objects under changing view positions, object tracking by means of image stabilization processing, view synthesis of 3D objects from 2D views of the same without recovering their 3D structure and many other applications.

Multiple view situations in Computer Vision have been analyzed with two objectives: to derive scene-independent constraints relating multiple views and to derive view-independent constraints relating multiple scene points. While the first approach seeks to model the configuration of cameras, the second attempts to characterize the configuration of points in the 3D world. The focus of our work is related to the second approach – to derive constraints on the configuration of the points being imaged in a manner that does not depend on the viewpoint or imaging parameters. Non-rigid motion is difficult to analyze in this scheme. The case of multiple objects moving with different velocities or accelerations can be considered very close to the case of non-rigid motion. We have developed constraints on the projections of such point configurations which can be categorized into two classes: constraints that are time-dependent – which are functions of time, and constraints that are time-independent – which hold at every time instant.

Bennet and Hoffman showed that polynomials to characterize a configuration of stationary points in a view-independent manner can be constructed from 2 views of 4 points under orthographic projection. This was extended by Carlsson to the case of scaled orthographic projection using 2 views of 5 points. Shashua and Levin generalized this to the case of an affine projection model for a time-dependent view-independent constraint on the projections of 5 points moving with different but constant velocities.

We show that the projection of a point moving with constant velocity in the world moves with constant velocity in the image when the camera is affine. This result is used to formulate a view and time-independent constraint on the velocities of the projections of 4 points whose computation needs 2 views. This is a significant theoretical contribution as it needs fewer points and accommodates a more general imaging model. In the same manner, points moving with constant acceleration in the world move with constant acceleration in the image as well. We derive time-dependent and time-independent constraints on respectively the velocities and accelerations of the projections of 4 points. These view-independent constraints can be used to recognize a configuration of 4 moving points and also to align frames of synchronized videos.

Though, points moving with independent uniform velocities or accelerations model many non-rigid motion conditions, it is desirable to have a technique that accommodates general non-linear motion. We make the observation that

as a point moves in the world it traces out a contour in the world and the trajectory traced out by its projection in an image would be the projection of the world contour. Thus, the contours traced out in different views would correspond. So the problem of analysing the non-rigid motion of a point can be transformed to the problem of analysing the contour traced out by its projections in various views. When the non-rigid motion in the world is restricted to a plane, that is when the motion is planar, the contours traced out in the views can be thought to be projections of a planar shape, the shape being the trajectory in the world. We discuss how planar shape recognition techniques can be used to recognize and analyse the contours.

We then combined the motion constraints with properties of a contour to devise a mechanism for recognizing a deforming contour, points on whose boundary move with independent uniform linear velocity or acceleration. Given two views of the deforming contour in a reference view, we can now recognize the same contour in any view at any time instant. We have also derived novel view-dependent parameterizations of the motion of the projections of points moving with uniform linear motion in the world, which enable us to arrive at view-independent constraints on the projections of points in motion that are simpler than the ones reported in literature.

Chapter 1

Introduction

Artificial Intelligence experts in the 1960s felt that it was only a matter of time before we would have machines that could see. That prophecy remains unfulfilled half a century later and would remain so for quite some more time. In course of our efforts to achieve that dream, a new discipline has emerged – Computer Vision, which encapsulates mathematics, computer science, human perception and biology. Computer Vision aims at extracting information from images much like the human brain does from the images captured by our eyes. Significant progress has been made in this field which is reflected in a number of applications in surveillance, industry, and the entertainment world.

Advances in Computer Vision have been helped by advances in Image Processing techniques [3, 4]. Image processing is the process of taking an input image and processing it, so that it attains certain desired characteristics. For instance, we use Image Processing techniques to increase the brightness of an image, to sharpen an image and so on. In a block diagram form, Image Processing techniques take as input an image and produce another image as output (Figure 1.1(a)). In contrast to this, Computer Vision techniques take as input an image or a set of images and produce *information* as output. (Figure 1.1(b)). Computer Vision is thus much more than Image Processing; Image Processing is only a step in the objective of Computer Vision. For example, in Geographical Information Systems (GIS), images captured by satellites are first 'Image Processed' to enhance them and then further processed so as to automatically extract the location, spread and other characteristics of features of interest like roads, rivers, vegetation, etc. Advances in Image Processing have enabled us to provide better inputs to image understanding algorithms thereby enabling us to correctly 'understand' more.

Computer Vision research spans the entire range of issues – the structure and properties of the world scene being imaged, the illumination in the scene, the properties and positioning of the imaging systems, the relationships between the images of the world scenes, and their use in applications like recognizing faces, shapes, textures , reconstructing the 3D world from image(s), creating views of how a scene would look from a novel viewpoint, etc. Theoretical studies have made a lot of ground especially in Geometric Computer Vision which deals with the analysis of relationships between the world and its image, as well as the constraints and properties of images of the world taken from multiple view points. An image is the projection of the 3D world onto the 2D plane of the camera. This projection results in the loss of information present in the third dimension, popularly referred to as the depth or the z dimension. (To see this for yourself hold your hands at different distances from your cameras (eyes) and see only through one eye. You would find it difficult to ascertain which hand is farther away; however you can easily do the same with two eyes!) It is easy to see that a plurality of projections (cameras) can compensate for this loss more than a single view and this has led to the study of the geometry that underlies multiple views of the same scene. These studies have brought to light a number of intricate and beautiful geometric relations that exist between images of objects in the world which have been used for a number of interesting and hitherto impossible applications.

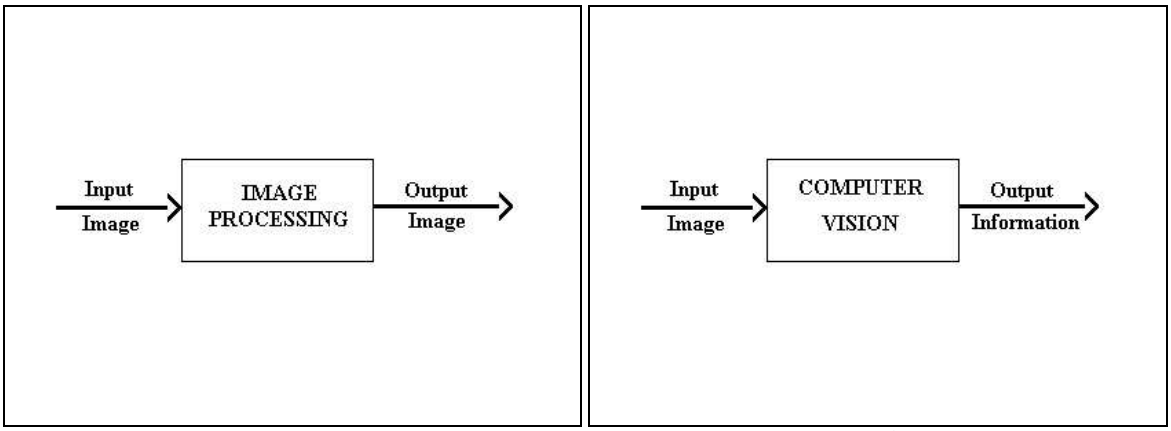

Figure 1.1: Block Diagrams for (a) Image Processing (b) Computer Vision

The geometric constraints on the projections of points were derived by Kruppa [5] in his studies in photogrammetry. The structure of points and lines as seen in two views attracted the attention of computer vision researchers like Longuet-Higgins [6] and Oliver Faugeras [7] in the eighties and early nineties. Similar studies on the underlying constraints in three views followed [8, 9, 10, 11, 12]. The structure of greater than three has also been studied [13, 10]. Two excellent textbooks have recently appeared on multiview geometry for computer vision [1, 14]. The mathematical structure underlying multiple views have been studied with respect to projective, affine, and Euclidean frameworks of the world with amazing results. These multi view relations have been used for visual recognition of objects under changing view positions, [15, 16, 17, 18], aligning frames of synchronized videos, [19, 20, 21, 22], synthesizing views from novel viewpoints of 3D objects from 2D views of the same without recovering their 3D structure [23, 24, 25, 26, 27], and 3D reconstruction from image sequences [28, 29, 30, 31] to name a few application areas.

Computer Vision applications discussed above are suited for static scenes (except for frame alignment) or when we have solitary images from different viewpoints. How about studying videos of dynamic scenes or imaging using a moving camera? There is tremendous redundancy in the information in video sequences as the information content does not change dramatically across frames, redundancy which can be exploited to solve a number of problems like motion and scene structure estimation [32, 33, 34], recognition of moving bodies [35, 36](add more) and tracking [37, 38, 39].

Views of the same shape or the same texture or the same world motion when taken from multiple view points and/or with cameras having different imaging parameters differ(Figure 1.2). Recognizing them to be the same is a challenging problem, which when solved would provide interesting handles to solve higher level vision problems like 3D reconstruction, camera motion estimation, etc. Thus, recognition is a preliminary step in image understanding. Recognition of shapes has been studied extensively for many years now. Approaches to achieve shape recognition include recognition by alignment [16], linear combination of models [18], polygonal approximation [40], and algorithms based on geometrically invariant features [2, 4]. Boundaries are also recognised by modeling them in a transform domain like the Fourier domain [41, 42]. In course of my undergraduate thesis we had formulated a novel shape recognition mechanism combining shape properties with multiview constraints in the Fourier domain [43, 44, 45, 46, 47].

Textures have been studied to develop recognition techniques that are invariant to illumination of the scene and pose of the cameras imaging the texture region [48, 49, 50]. As an application itself texture recognition is an important

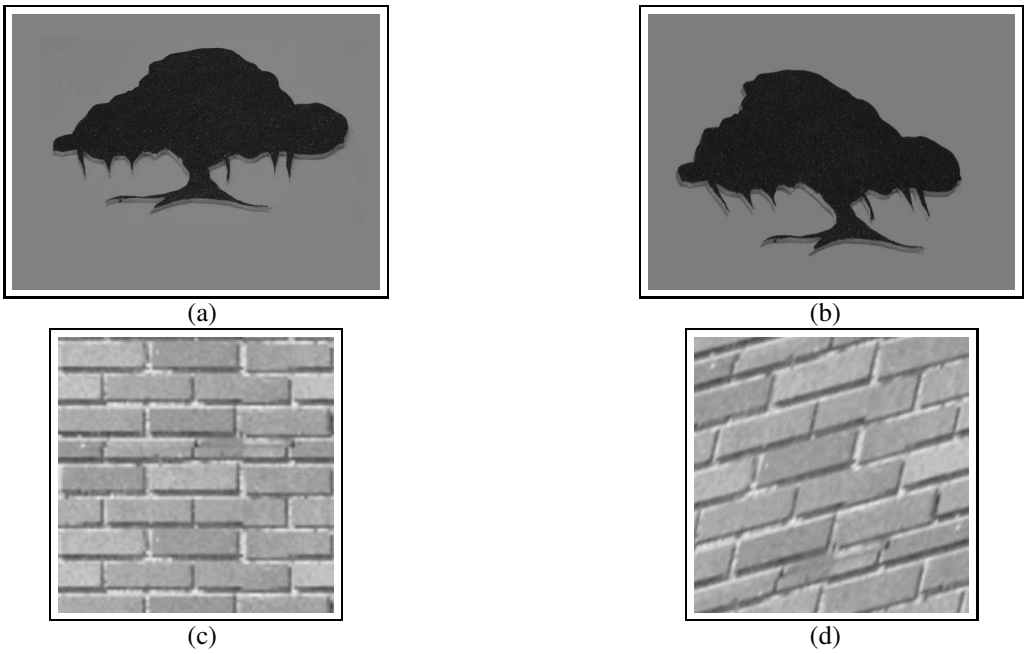

Figure 1.2: (a) and (b) : Two views of a shape. (c) and (d): Two views of a Texture

problem. It also plays an important role in many computer vision algorithms like in wide-baseline stereo [50].

Recognition of moving objects is still being investigated [35, 36]. In course of this work we have focused on deriving constraints on images of moving objects that can be used for applications like recognition and alignment of frames of synchronized videos.

To enable the reader to become familiar with the notions and concepts in the geometry of multiple views, a brief introduction is presented next.

1.1 Homogeneous Representation

Using homogeneous representation immensely simplifies expressions in geometry, hence their widespread use, particularly in Computer Vision.

1.1.1 Lines

A line on a 2D plane is defined by $ax + by + c = 0$. It can, therefore, be defined by the vector $\begin{bmatrix} a & b & c \end{bmatrix}^T$. However, we note that the line given by $kax + kby + kc = 0$ is the same line and hence the vectors $\begin{bmatrix} a & b & c \end{bmatrix}^T$ and $\begin{bmatrix} ka & kb & kc \end{bmatrix}^T$ are equivalent, provided $k \neq 0$. An equivalence class of vectors under this scaling equivalence relationship is known as a *homogeneous* vector. The set of equivalence class of vectors in $\mathbb{R}^3 - (0,0,0)^T$ form the *projective space* $\mathbb{P}^2$ [1].

1.1.2 Points

Given a line $\mathbf{l} = \begin{bmatrix} a & b & c \end{bmatrix}^T$, a point (x, y) lies on it, iff $ax + by + c = 0$. This can be written as the dot product $(x, y, 1).\mathbf{l} = 0$ where the point (x, y) in $\mathbb{R}^2$ is written as $\begin{bmatrix} x & y & 1 \end{bmatrix}^T$ – a point in projective space $\mathbb{P}^2$ by adding a final coordinate of 1. We note that the incidence condition can be written as $akx + bky + ck = 0$, whereby the point (x, y) in $\mathbb{R}^2$ can be represented by $\begin{bmatrix} kx & ky & k \end{bmatrix}^T$, for all $k \neq 0$. Thus, points can also be expressed as homogeneous vectors. An arbitrary homogeneous vector $(x1, x2, x3)^T$ represents the point $(x1/x3, x2/x3)$ in $\mathbb{R}^2$. The representation $(x1/x3, x2/x3)$ is known as the inhomogeneous representation. A point with $x3 = 0$ is said to be a point at infinity. These points exist only in $\mathbb{P}^2$ space and not in Euclidean $\mathbb{R}^2$ space.

This homogeneous representation gives us simple expressions for a number of geometric conditions. For instance, when the point $\mathbf{x}$ lies on the line $\mathbf{l}$, we have $\mathbf{x}^T\mathbf{l} = 0$; the point of intersection of two lines $\mathbf{l}_1$ and $\mathbf{l}_2$ is given by $\mathbf{x} = \mathbf{l}_1 \times \mathbf{l}_2$; the line joining two points $\mathbf{x}_1$ and $\mathbf{x}_2$ is given by $\mathbf{l} = \mathbf{x}_2 \times \mathbf{x}_2$. Homogeneous representations for points and lines are used extensively in the discussions in the rest of this report.

1.2 Image-to-Image Homographies for Images of 2D Planes

When a planar scene is imaged from multiple view points or when a scene is imaged by cameras having the same optical centre, the images are related by homographies [1]. A homography or a collineation is a mapping from one plane to another such that the collinearity of any set of points is preserved. In other words a homography is an invertible mapping h from $\mathbb{P}^2$ to itself such that three points x_1, x_2 and x_3 lie on the same line if and only if $h(x_1)$, $h(x_2)$ and $h(x_3)$ do.

Plane-to-plane homographies can be categorised into isometry, similarity, affine and projective [1]. The later classes subsume the earlier ones i.e. isometry $\subset$ similarity $\subset$ affine $\subset$ projective.

Isometry : An Isometry is a transformation of the plane $\mathbb{R}^2$ that preserves Euclidean distance. Such a transformation is represented as

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \epsilon\cos\theta & -\sin\theta & t_x \\ \epsilon\sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

where $\epsilon = \pm 1$. If $\epsilon = 1$ then the isometry is orientation preserving and is a Euclidean Transformation. If $\epsilon = -1$ then the isometry reverses orientation and involves a reflection. The above can be expressed more compactly as $\mathbf{x}' = \mathbf{H_E}\mathbf{x}$ where $\mathbf{H_E} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0^T & 1 \end{bmatrix}$

where $\mathbf{R}$ is a 2×2 orthonormal rotation matrix and $\mathbf{t}$ is a translational 2-vector. An Euclidean transformation has three degrees of freedom, one for $\mathbf{R}$ and two for $\mathbf{t}$.

Similarity : A similarity transformation is an isometry with isotropic scaling. Such a transformation can be written as

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

or more compactly $\mathbf{x}' = \mathbf{H_S x}$, where $\mathbf{H_S} = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ 0^T & 1 \end{bmatrix}$ and s is the isotropic scaling factor. A similarity transformation is also known as an equi-form transformation as its preserves the shape form. Similarity transformations have four degrees of freedom, the additional degree of freedom over Euclidean transformations coming from the isotropic scale factor.

Affine : An affine transformation is a non-singular linear transformation followed by a linear translation. In the form of a matrix it can be represented as

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

or more compactly $\mathbf{x}' = \mathbf{H_A x}$ where $\mathbf{H_A} = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ 0^T & 1 \end{bmatrix}$ and $\mathbf{A}$ is a non-singular 2×2 matrix. Affine transformations have six degrees of freedom - four from $\mathbf{A}$ and two from $\mathbf{t}$.

Projective : A projective transformation is a general non-singular linear transformation of homogeneous coordinates. This generalizes an affine transformation, which is the composition of a general non-singular linear transformation of inhomogeneous coordinates and a translation.

A projective transformation can be expressed as

$$\begin{aligned} \mathbf{x}' &= \mathbf{H}_P \mathbf{x} \\ \mathbf{H}_P &= \begin{bmatrix} \mathbf{A} & t \\ \mathbf{V}^T & v \end{bmatrix} \text{ and } \mathbf{V} \text{ is a vector } [v_1, v_2] \end{aligned} \tag{1.1}$$

The vector $\mathbf{V}$ is the projective component of the homography and makes the transformation non-linear in inhomogeneous coordinates. A projective transformation has eight degrees of freedom – four for $\mathbf{A}$, two each for $\mathbf{t}$ and $\mathbf{V}$.

The image-to-image homography between two views of a scene is a general projective transformation when

- when the object being imaged is planar, or
- when the scene is imaged with cameras having the same optical centre.

Further restrictions on the imaging conditions/parameters give us more specific homographies like Euclidean, similarity and affine. Figure 1.3 shows various views of a hexagon under different image-to-image homographies. View (a) is the reference view from which other views were generated using appropriate homographies. Views (a) and (b) are related by isometric homographies, (c) and (d) by similarity transformations, (e) and (f) by affine homographies, while general projective homographies relate views (g) and (h). It can be seen that all lengths and angles are preserved in the views related by isometries. The hexagons in the views related by similarity transforms look similar (hence the name similarity) with all angles preserved; lengths however are not preserved. In the views related by affine homographies, neither lengths nor angles are preserved, but parallelism is maintained. While, in the views related by projective transformations none of lengths, angles and parallelism are maintained.

Table 1.1 summarizes the various homographies, their invariant geometric properties and scenarios when they are applicable. Transformations lower in the table are specializations of those above and inherit their invariants.

We have seen the form of various transformations in 2D space. When analysing transformations, many a times we look for properties that do not change under those transformations. So next we look at the properties that do not change under the transformations in 2D, i.e. invariants under 2D plane transformations.

Homography	Invariant Properties	Relevant Scenarios
Projective (8 dof)	Concurrency, collinearity, order of contact : intersection (1pt contact); tangency (2pt contact); inflections (3pt contact with line); tangent discontinuities and cusps. Cross ratio (ratio of ratio of lengths)	Imaging a planar object from multiple view points, imaging a scene with cameras having the same optical centre
Affine (6 dof)	Parallelism, ratio of areas, ratio of lengths on collinear or parallel lines (eg. midpoints), linear combinations of vectors(eg. centroids). The line at infinity.	Imaging a planar object with high focal length cameras, imaging a scene from a distance.
Similarity (4 dof)	Ratio of lengths, angle. Circular points	Imaging with different focal lengths
Euclidean (3 dof)	Length, area	Optical character readers

Table 1.1: Homographies, their invariant geometric properties and scenarios where they are relevant. Adapted from [1, 2]. Homographies lower in the table inherit the invariants of the homographies above them. (dof=degrees of freedom)

1.3 Invariants

The study of invariants has been pursued actively for many years. Invariants provide us with the ability to come up with representations of the features in a scene that do not depend on the view, and can prove to be extremely handy when processing information from multiple views.

Let p be a parameter vector subject to the linear transformation $\mathbf{H}^l$ to give p^l. An invariant $I(\cdot)$ defined on the geometric structure described by p should transformed according to $I(p^l) = I(p)|\mathbf{H}^l|^w$, where $|\mathbf{H}^l|$ is the determinant of the transformation [2]. w is the weight of the invariant. The invariant is said to be scalar when $w = 0$. Depending on the transformation the invariants vary. For instance, when the transformation is Euclidean, distances and angles are invariant, while in similarity transformations distances are no longer invariant though angles are. If the transformation $\mathbf{H}^l$ is affine, the invariants are called affine invariants. The ratio of areas and ratio of lengths on parallel lines are invariant to affine transformations. Several cross ratios are invariant under general projective transformations some of which are given next.

Cross-ratio of areas of five points: The cross-ratio of the areas of five points $\mathbf{x_1}$, $\mathbf{x_2}$, $\mathbf{x_3}$, $\mathbf{x_4}$, and $\mathbf{x_5}$, no three of which are collinear, is defined by

$$cr(\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, \mathbf{x_4}, \mathbf{x_5}) = \frac{\triangle_{\mathbf{x_1 x_2 x_5}} \cdot \triangle_{\mathbf{x_3 x_4 x_5}}}{\triangle_{\mathbf{x_1 x_3 x_5}} \cdot \triangle_{\mathbf{x_2 x_4 x_5}}} \ , \tag{1.2}$$

where $\triangle_{\mathbf{x_i x_j x_k}}$ is the area of the triangle formed by points $\mathbf{x_i}, \mathbf{x_j}, \mathbf{x_k}$. This is invariant to general linear or projective transformations [2].

Cross-ratio of four concurrent lines: The cross-ratio of four concurrent lines l_1, l_2, l_3, l_4 is defined as

$$cr(l_1, l_2, l_3, l_4) = \frac{sin\theta_{13} \cdot sin\theta_{24}}{sin\theta_{23} \cdot sin\theta_{14}}, \tag{1.3}$$

where θ_{ij} represents the angle formed by the lines l_i and l_j. This is also invariant to a general projective transformation [2].

1.3.1 Number of Invariants

Given an invariant $I(\cdot)$, for every k, $k * I(\cdot)$ is also invariant. Similarly, if $I_1(\cdot)$ and $I_2(\cdot)$ are invariant, so are $c_1 * I_1(\cdot) + c_2 * I_2(\cdot)$, where c_1 and c_2 are constants. So the number of invariants are potentially infinite. However, the number of functionally independent invariants are not. If there is a configuration space S on which a transformation H acts, then the number of functionally independent scalar invariants is $\geq$ (dim S - dim H) [2]. Thus, if we have a configuration of 5 points, then there exist atleast 2 functionally independent scalar invariants under a general projective transformation.

Till now, we have restricted ourselves to the transformations in the 2D plane. In the next Section, we look at transformations in 3D. As we will see they are simple extensions to the transformations of the 2D plane.

1.4 Projective Transformations in 3D

In this Section, we explore the transformations of 3D projective space. A projective transformation in 3D is of the form.

$$\mathbf{x}' = \mathbf{H}\mathbf{x}$$

where $\mathbf{x}$ and $\mathbf{x}'$ are 4-vectors and $\mathbf{H}$ is 4×4. Since the transformation is in homogeneous coordinates, the number of degrees of freedom of the transformation matrix is 15. Analogous to the hierarchy of projective transformations in 2D, we have a similar hierarchy of transformations in 3D.

Euclidean An Euclidean transformation preserves Euclidean distance. Such a transformation is represented as

$$\mathbf{H}_E = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

where $\mathbf{R}$ is a 3×3 orthonormal rotation matrix and $\mathbf{t}$ is a translational 3-vector. This transformation has six degrees of freedom – three each for $\mathbf{R}$ and $\mathbf{t}$.

Similarity : A similarity transformation combines a Euclidean transformation with isotropic scaling. Such a transformation can be written as

$$\mathbf{H}_S = \begin{bmatrix} s\mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

where s is the isotropic scaling factor. This transformation has seven degrees of freedom, the isotropic scale factor s contributing an additional degree to that of the Euclidean transformation.

Affine : An affine transformation is a non-singular linear transformation of inhomogeneous coordinates followed by a linear translation. It can be represented as

$$\mathbf{H}_A = \begin{bmatrix} \mathbf{A} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

and $\mathbf{A}$ is a non-singular 3×3 matrix. The transformation has twelve degrees of freedom, nine for $\mathbf{A}$ and three for $\mathbf{t}$.

Projective : A projective transformation is a general non-singular linear transformation of homogeneous coordinates and can be expressed as

$$\mathbf{H}_P = \begin{bmatrix} \mathbf{A} & t \\ \mathbf{V}^T & v \end{bmatrix}$$

This transformation has fifteen degrees of freedom, the three additional degrees over an affine transformation coming from $\mathbf{V}$.

1.5 Imaging

Imaging is the process in course of which a 3D world is projected onto the 2D plane of the camera to result in what we call an 'image' of the world. Thus, the process of imaging is a transformation from 3D to 2D which depends on the characteristics of the camera and the relative orientations of the world scene and the imaging camera. This camera mapping transformation is characterized by matrices. In this section, we discuss two kinds of cameras – cameras whose centre is finite, and those whose centre is at infinity.

1.5.1 Finite Cameras

The pinhole camera is the simplest finite camera model whose operation is shown in figure 1.4. The camera centre (also known as the optical centre) is at the origin of the coordinate system, with the image being formed on the plane $z = f$ which is referred to as the focal or image plane. A point in space with coordinates $\mathbf{X} = \begin{bmatrix} X & Y & Z \end{bmatrix}^T$ is imaged at the point $\mathbf{x}$ on the image plane, where the line joining $\mathbf{X}$ and the camera centre $\mathbf{C}$ intersects the image plane. The principal axis is the line from the camera centre perpendicular to the image plane, its intersection with the image plane being the principal point. The plane through the camera centre parallel to the image plane is called the principal plane of the camera. It can be seen that

$$\mathbf{x} = \begin{bmatrix} fX/Z & fY/Z \end{bmatrix}^T$$

Representing the world and image points using homogeneous coordinates, the projection from $\mathbb{R}^3$ to $\mathbb{R}^2$ can be expressed as a linear mapping between the homogeneous coordinates.

$$\mathbf{x} \equiv \begin{bmatrix} fX \\ fY \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

This can be written as

$$\mathbf{x} = \mathbf{M}\mathbf{X} \tag{1.4}$$

where $\mathbf{M} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$ is the camera matrix for the pinhole camera. Thus transformation of the 3D world points by a 3×4 camera matrix results in the projection of the world point onto the image plane.

Internal Parameters When the origin of the image plane does not coincide with the principal point (p_x, p_y), the projection matrix becomes

$$\begin{aligned} \mathbf{M} &= \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \\ &= \mathbf{K}[\mathbf{I}|\mathbf{0}] \\ \text{where } \mathbf{K} &= \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

represents the internal parameters of the camera and is known as the camera calibration matrix. In cameras like CCD cameras, pixels are non-square resulting in unequal scale factors in each direction. The calibration matrix now becomes

$$\mathbf{K} = \begin{bmatrix} \alpha_x & 0 & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

where $\alpha_x = fm_x$ and $\alpha_y = fm_y$ represent the focal lengths in the x and y directions in terms of pixel dimensions and $x_0 = m_x p_x$ and $y_0 = m_y p_y$.

To accommodate the case of the x and y directions not being perpendicular we introduce a skew parameter (s) in $\mathbf{K}$. $\mathbf{K}$ now becomes

$$\mathbf{K} = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

The skew parameter is usually zero for most cameras. Note that the internal calibration matrix is upper triangular.

External Parameters The above camera model has been derived when the camera coordinate frame coincides with the world coordinate frame. When this is not the case, we need a 3D rotation $(\mathbf{R})$ and a translation to map a point in the world coordinate frame to the camera coordinate frame. This rotation and translation constitute the external parameters of the camera. Let $\tilde{\mathbf{C}}$ be the inhomogeneous representation of the camera centre in the world coordinate frame, then the required transformation in homogeneous coordinates becomes

$$\mathbf{X}' = \begin{bmatrix} \mathbf{R} & -\mathbf{R}\tilde{\mathbf{C}} \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{X}$$

Therefore, the projection is given by

$$\mathbf{x} = \mathbf{K}\mathbf{R}[\mathbf{I}| - \tilde{\mathbf{C}}]\mathbf{X} \tag{1.5}$$

The number of degrees of freedom of this projection matrix is 11 ; 5 of $\mathbf{K}$, 3 of $\mathbf{R}$ and 3 of $\tilde{\mathbf{C}}$. Note that this is same number of degrees of freedom that we would get if we consider $\mathbf{M}$ to be a general 3×4 matrix since the transformation is of homogeneous coordinates where overall scale is unimportant.

A general finite projective camera is therefore given by

$$\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix}$$

A Projective Space Interpretation The external parameters of the camera align the world coordinate frame with the camera coordinate frame. We can think of this transformation as a 4×4 homography in $\mathbf{P}^3$ space. The internal parameters of the camera can be thought of the 3×3 homography that maps the image plane of the camera to the ideal image plane (at $z = f$ and where principal point coincides with the image origin). Thus, we can decompose the projective camera as

$$\mathbf{M} = [3 \times 3 \text{ homography}] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} [4 \times 4 \text{ homography}]$$

1.5.2 Cameras at Infinity

This class of cameras include cameras whose centre lies on the plane at infinity. For such cameras the left 3×3 minor of the camera matrix $\mathbf{M}$ is singular. There are two kinds of cameras at infinity – affine cameras and non-affine cameras.

Affine Cameras

Affine cameras are cameras whose projection matrix has $(0, 0, 0, 1)$ as the last row. It maps points at infinity to points at infinity. As above, we develop the affine camera from the most constrained case to the most general.

Orthographic Projection Orthographic projection of a world point $\begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T$ along the z-axis results in the image of the world point being formed at $\begin{bmatrix} X & Y & 1 \end{bmatrix}^T$. This transformation is represented by

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For a general orthographic projection, to accommodate for the difference in the camera coordinate frame and the world coordinate frame, we post multiply the projection matrix with $\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$ to get the orthographic camera

$$\mathbf{M} = \begin{bmatrix} \mathbf{r}^1 & \mathbf{t}^1 \\ \mathbf{r}^1 & \mathbf{t}^2 \\ \mathbf{0}^T & 1 \end{bmatrix}$$

where $\mathbf{r}^i$ and $\mathbf{t}^i$ represent the ith row of $\mathbf{R}$ and $\mathbf{t}$ respectively. The orthographic camera has five degrees of freedom – three from $\mathbf{R}$ and one each from $\mathbf{t}^1$ and $\mathbf{t}^2$.

Scaled Orthographic Projection Scaled orthographic projection is orthographic projection followed by isotropic scaling. Thus, the projection matrix is of the form

$$\begin{aligned}\mathbf{M} &= \begin{bmatrix} k & 0 & 0 \\ 0 & k & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{r}^1 & \mathbf{t}^1 \\ \mathbf{r}^1 & \mathbf{t}^2 \\ \mathbf{0}^T & 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{r}^1 & \mathbf{t}^1 \\ \mathbf{r}^1 & \mathbf{t}^2 \\ \mathbf{0}^T & 1/k \end{bmatrix} \end{aligned} \tag{1.6}$$

This projection matrix has six degrees of freedom, the additional degree of freedom over orthographic projection coming from the isotropic scale factor.

Weak Perspective Projection In scaled orthographic we have considered isotropic scaling. When the scaling is unequal along the two axes we get a weak perspective camera. The projection matrix of this camera is of the form

$$\mathbf{M} = \begin{bmatrix} \alpha_x & 0 & 0 \\ 0 & \alpha_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{r}^1 & \mathbf{t}^1 \\ \mathbf{r}^1 & \mathbf{t}^2 \\ \mathbf{0}^T & 1 \end{bmatrix}$$

This camera has eight degrees of freedom, the additional degrees of freedom over orthographic projection coming from the anisotropic scale factors along the two axial image directions.

General Affine Camera The above affine cameras have restrictions on their elements. For instance, the top two rows of the projection matrix are orthonormal and of unit norm for orthographic cameras; are orthonormal and of the same norm for scaled orthographic projection; and are orthonormal for weak perspective projection. The general affine camera has no constraints on its elements. It is a general 3×4 matrix of the form

$$\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{m_1} & m_{14} \\ \mathbf{m_2} & m_{24} \\ \mathbf{0} & 1 \end{bmatrix} \tag{1.7}$$

This matrix has 8 degrees of freedom. The only restriction on the affine camera is that since the rank of $\mathbf{M}$ is 3, the rank of the upper left 2×3 minor must be 2.

A Projective Space Interpretation In the case of projective cameras, we had argued that the projective camera matrix is a composition of a projective transformation in 3D, a projection from 3D to 2D followed by a projective transformation in 2D. In a similar manner, an affine camera matrix is a composition of an affine transformation in 3D, an orthographic projection from 3D to 2D along the z-axis followed by an affine transformation in 2D.

$$\mathbf{M} = [3 \times 3 \text{ affine homography}] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} [4 \times 4 \text{ affine homography}]$$

Affine∼Projective Projective cameras are the most general cameras. However, affine imaging can approximate projective imaging when the points in the scene do not differ much in depth and when the points being imaged are not far from the principal ray. In general, cameras with high focal length lenses satisfy both these constraints and so such imaging systems can be modeled as affine cameras. Infact many situations in multiview imaging can be approximated by affine imaging models [51].

Non-Affine Cameras at Infinity

For affine cameras the camera centre lies at infinity and the principal plane is the plane at infinity. However, it is possible for the principal plane of a camera to not be the plane at infinity and still have the camera centre at infinity. Such (rare) cameras do not have the last row as $(0, 0, 0, 1)$ as affine cameras and hence are non-affine. However it is still classified as a camera at infinity as its camera centre lies on the plane at infinity.

1.6 Multiview Constraints

1.6.1 Fundamental Matrix

The fundamental matrix [1] encodes a linear, epipolar constraint between projections of the same point in two views. If we have two snap shots of the same scene from two different viewpoints then $\exists$ a 3 x 3 matrix of maximum rank 2 called the Fundamental Matrix that relates corresponding points in the two views. The Fundamental Matrix constrains the location of the corresponding point to lie on a line in the other view.

Mathematically

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

where, $\mathbf{x}$ is a point in the first image, $\mathbf{x}'$ is the point in the second image that corresponds to $\mathbf{x}$, and $\mathbf{F}$ is the Fundamental Matrix.

The Fundamental matrix has 7 degrees of freedom. Since this definition is in homogeneous coordinates the Fundamental Matrix has 8 unknowns upto scale. One degree of freedom is then accounted for by the fact that the matrix is singular (has rank 2).

When the cameras are affine, the upper 2×2 matrix becomes $\mathbf{0}_{2\times 2}$

1.6.2 Trilinear Tensor

The Tri-Linear Tensor is to three views what the Fundamental Matrix is to two, constraining where the image of a point lies in a third view, given its position in two views. It encapsulates all the projective geometric relations between three views, that are independent of the scene structure. The Tri-Linear Tensor is a relationship of either of

- line-line-line correspondence,
- point-line-line correspondence,
- point-line-point correspondence,
- point-point-line correspondence,
- point-point-point correspondence

between the three views. The tensor only depends on the motion between views and the internal parameters of the cameras and is defined uniquely by the camera matrices of the views. This tensor, however, can also be computed by making use of image correspondences alone, without knowledge of the motion or calibration.

Point-Point-Point correspondence Let $\mathbf{P}$ be a point in 3D space that is projected onto three views at image locations $\mathbf{x}^1(x^1, y^1, 1)$, $\mathbf{x}^2(x^2, y^2, 1)$ and $\mathbf{x}^3(x^3, y^3, 1)$ respectively. Then there are four trilinear equations among the projections in the three views which are of the form

$$x^3\mathcal{T}_i^{13}\mathbf{x}^{1^i} - x^3x^2\mathcal{T}_i^{33}\mathbf{x}^{1^i} + x^2\mathcal{T}_i^{31}\mathbf{x}^{1^i} - \mathcal{T}_i^{11}\mathbf{x}^{1^i} = 0 \tag{1.8}$$

$$y^3\mathcal{T}_i^{13}\mathbf{x}^{1^i} - y^3x^2\mathcal{T}_i^{33}\mathbf{x}^{1^i} + x^2\mathcal{T}_i^{32}\mathbf{x}^{1^i} - \mathcal{T}_i^{12}\mathbf{x}^{1^i} = 0 \tag{1.9}$$

$$x^3\mathcal{T}_i^{23}\mathbf{x}^{1^i} - x^3y^2\mathcal{T}_i^{33}\mathbf{x}^{1^i} + y^2\mathcal{T}_i^{31}\mathbf{x}^{1^i} - \mathcal{T}_i^{21}\mathbf{x}^{1^i} = 0 \tag{1.10}$$

$$y^3\mathcal{T}_i^{23}\mathbf{x}^{1^i} - y^3y^2\mathcal{T}_i^{33}\mathbf{x}^{1^i} + y^2\mathcal{T}_i^{32}\mathbf{x}^{1^i} - \mathcal{T}_i^{22}\mathbf{x}^{1^i} = 0 \tag{1.11}$$

where $\mathcal{T}_k^{ij}$ is the $3 \times 3 \times 3$ Trilinear Tensor. Every corresponding triplet $\mathbf{x}^1$, $\mathbf{x}^2$, $\mathbf{x}^3$ contributes four linearly independent equations and the number of unknowns in the tensor is 27 (26 upto scale). Therefore, a minimum of seven corresponding points across the three views are needed to determine the Tri-Linear Tensor (up to a scale).

Line-Line-Line correspondence Let L be a 3D line that is imaged as l^1, l^2, and l^3 in three views. Then the Trilinear Tensor, $\mathcal{T}$, relates the lines as

$$l^1 = (l^2)^{\mathrm{T}}\mathcal{T}l^3 \tag{1.12}$$

Thus, the tensor can be used to transfer a line into a view given its location in two other views.

1.7 Motion in Multiple Views

Over the past half a century, a lot of work has been done on analysing and understanding solitary images taken from multiple viewpoints. A logical extension to this is analysing a stream of images of a dynamic scene. Such an analysis would be of immense use as most scenes of interest are dynamic and not static. If we have a configuration of points in the world, whose motion has certain structure (say they move with independent uniform velocities or accelerations) then when they are imaged by a camera (which too has a definite structure) one would observe a structured motion of the projections of the world points in the images. Given the world structure, the structure of the camera, and their parameters, we can determine the structure and parameters of the motion in the images. Can we determine constraints on the parameters of the motion of the projections when we know only the structure but not the parameters of the world motion and the camera? This question is answered in Chapter 2 wherein we derive a number of linear view-independent constraints on the projections of points moving with independent uniform velocities or accelerations. Non-rigid motion of configurations is difficult to model and study. Though, points moving with independent uniform velocities or accelerations model many non-rigid motion conditions, it is desirable to have a technique that accommodates general non-linear motion. We make the observation that as a point moves in the world it traces out a contour in the world and the trajectory traced out by its projection in an image would be the projection of the world contour. Thus, the contours traced out in different views would correspond. So the problem of analysing the non-rigid motion of a point can be transformed to the problem of analysing the contour traced out in the various views. When the non-rigid motion in the world is restricted to a plane, that is when the motion is planar, the contours traced out in the views can be thought to be projections of a planar shape, the shape being the trajectory in the world. A number of planar shape recognition techniques have been developed which can be used to recognize and hence analyse the contours. This approach is outlined in Chapter 3. Chapter 2 analyses the motion of the projection of a point undergoing structured motion. A question that comes to mind is if we have shape and structured motion constraints can we recognize a deforming contour, when the deformation has some structure? This question is answered and a novel recognition mechanism is presented in Chapter 4. The constraints and analysis presented in Chapters 2, 3, and 4 deal

with the scenario when the camera model is affine. Motion under projective camera models is presented in Chapter 5. Applications of the constraints detailed in this report to the alignment of frames of synchronized videos is presented in Chapter 6. In Chapter 7, a few concluding remarks are presented.

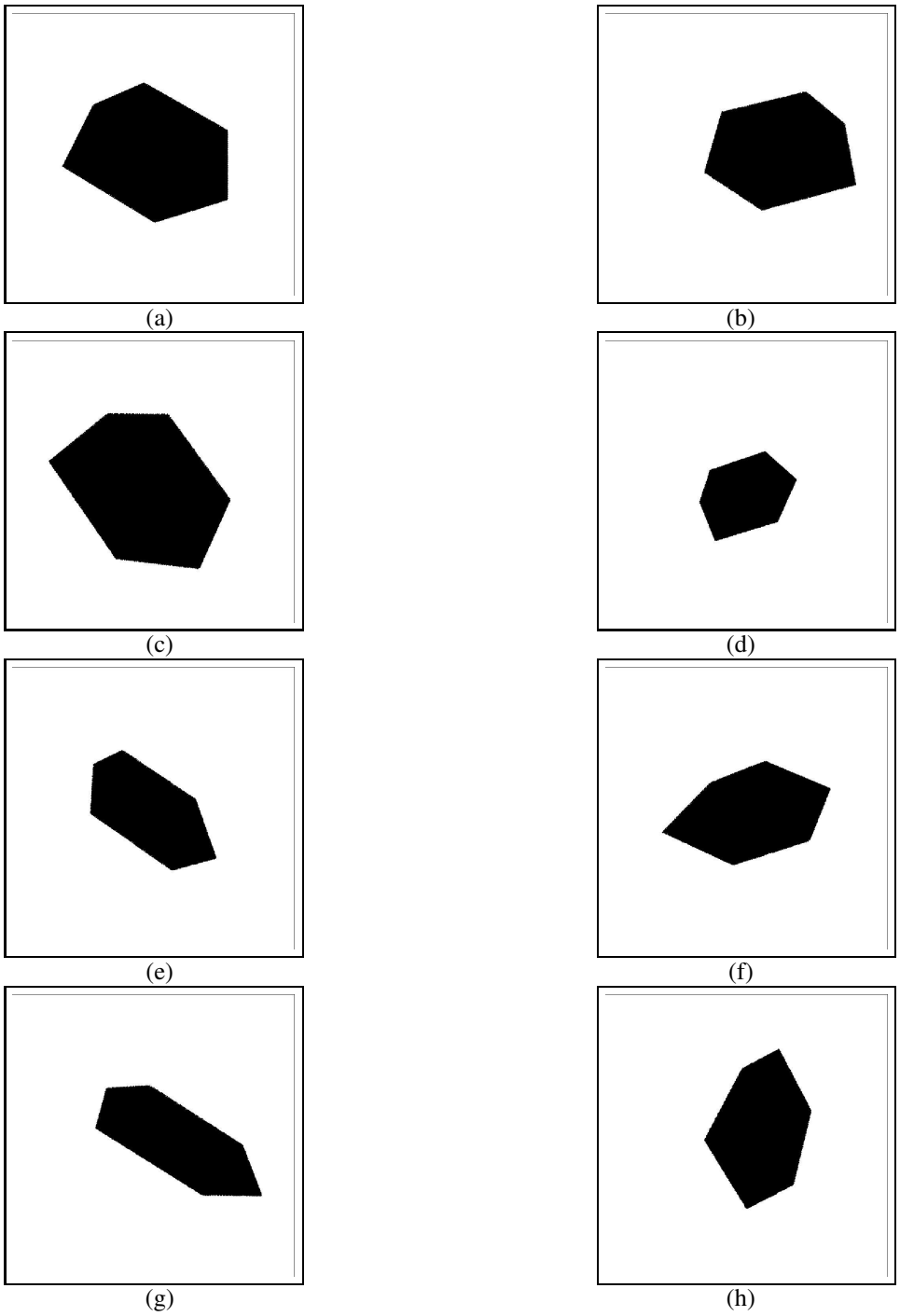

Figure 1.3: Several views of a hexagon for different image-to-image homographies

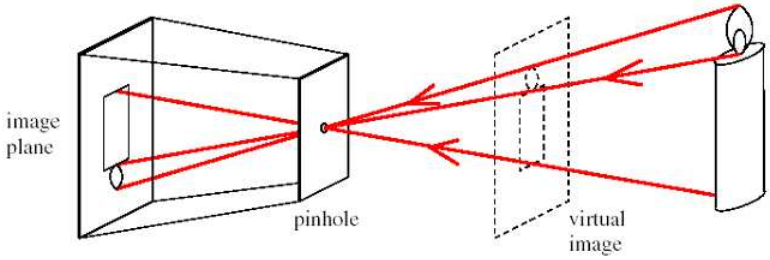

Figure 1.4: Imaging using a Pinhole camera

Chapter 2

View Independent Constraints on Point Configurations

The study of view independent constraints on the projections of a configuration of points is an area of active research today as they provide a convenient cue for recognition of such point configurations. A number of view independent features (also known as invariants) have been identified for static point configurations [1, 2]. We use a generic approach to deriving linear view independent constraints for static point configurations. We factor out the camera parameters by considering a configuration of points, thus achieving independence to the camera and becoming view independent – valid in all views of the same configuration. This principle is then used for deriving novel view independent constraints on the affine projections of moving points. Some of the constraints for configurations in motion are time-independent i.e. they are applicable at all times in all views, while others are time-dependent and are applicable only for proper choices of the time parameter in the constraint.

Organisation of the chapter In Section 1, we investigate the view independent constraints on a static configuration of points. Starting from the most general configuration of points we consider special cases of the points being on a 3D line, on general 3D lines, on concurrent 3D lines, on a plane, on a line on a plane, on general lines on a plane and on concurrent lines on a plane and derive the computational requirements for the determining the view independent constraints for such configurations. Non-rigid motion is difficult to analyze. However, the case of multiple objects moving with different uniform velocities or accelerations can be considered very close to the case of non-rigid motion. These cases are explored in Section 2 where we derive time-independent and time-dependent view independent constraints on the projections of a configuration of points moving with independent uniform velocities or accelerations. Results of experiments conducted to verify these constraints are also presented.

2.1 Static Point Configurations

The study of invariants has been pursued actively in Computer Vision for many years [1, 2]. These studies aim at computing values from the projections of point configurations that are the same for all views of the configuration. In this Section, we derive and analyse view-independent constraints that the projections of static configurations must satisfy taking into account all possible configuration types.

Let $\mathbf{P}_i = \begin{bmatrix} P_{ix} & P_{iy} & P_{iz} \end{bmatrix}^T$, $i = 1...5$ be a configuration of static points in the world.

Let $\mathbf{p}_i = \begin{bmatrix} x_i & y_i & 1 \end{bmatrix}^T$ be the projection of $\mathbf{P}_i$ when viewed by an affine camera $\mathbf{M} = \begin{bmatrix} \mathbf{m}_1 & m_{14} \\ \mathbf{m}_2 & m_{24} \\ \mathbf{0} & 1 \end{bmatrix}$ where $\mathbf{m}_i$ is the vector of the first 3 elements in the ith row of $\mathbf{M}$ (Equation 1.7). Therefore,

$$x_i = \mathbf{m}_1.\mathbf{P}_i + m_{14}$$
$$y_i = \mathbf{m}_2.\mathbf{P}_i + m_{24}$$

or alternatively

$$\begin{bmatrix} P_{ix} & P_{iy} & P_{iz} & 1 & x_i \end{bmatrix} . \begin{bmatrix} \mathbf{m}_1 & m_{14} & -1 \end{bmatrix}^T = 0$$
$$\begin{bmatrix} P_{ix} & P_{iy} & P_{iz} & 1 & y_i \end{bmatrix} . \begin{bmatrix} \mathbf{m}_2 & m_{24} & -1 \end{bmatrix}^T = 0$$

If we have five points, then we can form matrices

$$\mathbf{C}_x = \begin{bmatrix} P_{1x} & P_{1y} & P_{1z} & 1 & x_1 \\ P_{2x} & P_{2y} & P_{2z} & 1 & x_2 \\ P_{3x} & P_{3y} & P_{3z} & 1 & x_3 \\ P_{4x} & P_{4y} & P_{4z} & 1 & x_4 \\ P_{5x} & P_{5y} & P_{5z} & 1 & x_5 \end{bmatrix}$$
$$\mathbf{C}_y = \begin{bmatrix} P_{1x} & P_{1y} & P_{1z} & 1 & y_1 \\ P_{2x} & P_{2y} & P_{2z} & 1 & y_2 \\ P_{3x} & P_{3y} & P_{3z} & 1 & y_3 \\ P_{4x} & P_{4y} & P_{4z} & 1 & y_4 \\ P_{5x} & P_{5y} & P_{5z} & 1 & y_5 \end{bmatrix} \tag{2.1}$$

such that $\mathbf{C}_x \begin{bmatrix} \mathbf{m}_1 & m_{14} & -1 \end{bmatrix}^T = 0$, and $\mathbf{C}_y \begin{bmatrix} \mathbf{m}_2 & m_{24} & -1 \end{bmatrix}^T = 0$. Therefore $\mathbf{C}_x$ and $\mathbf{C}_y$ are rank deficient, i.e. the ranks of $\mathbf{C}_x$ and $\mathbf{C}_y$ are at most 4, implying $|\mathbf{C}_x| = |\mathbf{C}_y| = 0$.

2.1.1 General 3D Configurations of points

When the points $\mathbf{P}_i$ are in general configuration, the rank of matrices $\mathbf{C}_x$ and $\mathbf{C}_y$ is 4. On using $|\mathbf{C}_x| = 0$ and $|\mathbf{C}_y| = 0$, we get

$$\sum_i^5 \alpha_i x_i = 0$$
$$\sum_i^5 \alpha_i y_i = 0 \tag{2.2}$$

where α_i are functions of the world position of the points $\mathbf{P}_i$ and hence is the same for all views, i.e the αs are view-independent coefficients. Note that the coefficients of x_i and y_i in the above constraint are the same αs. Thus, the total number of unknowns is 4 (upto scale). Each view gives two equations in terms of α. Therefore, we need two views of the five points to compute all the view-independent coefficients.

Lemma 1 We need two views of five points in general configuration to obtain a linear view independent constraint on the projections of the five points.

This constraint holds for all views of the same general 3D configuration. Next we look at the simplifications that arise when the point configuration is not general.

2.1.2 Points on a 3D Line

Every point on a 3D Line can be expressed in terms of two other points on that line. If we are trying to characterize a configuration of points on a 3D line, then the rank of $\mathbf{C}_x$ and $\mathbf{C}_y$, would be at most 2. Therefore any of their 3×3 minors would be rank deficient.

$$\left|\begin{bmatrix} P_{1x} & 1 & x_1 \\ P_{2x} & 1 & x_2 \\ P_{3x} & 1 & x_3 \end{bmatrix}\right| = \left|\begin{bmatrix} P_{1x} & 1 & y_1 \\ P_{2x} & 1 & y_2 \\ P_{3x} & 1 & y_3 \end{bmatrix}\right| = 0$$

which give the constraint

$$\sum_i^3 \alpha_i x_i = 0$$
$$\sum_i^3 \alpha_i y_i = 0$$

where α_i are functions of the world position of the points $\mathbf{P}_i$ and hence is the same for all views, i.e the αs are view-independent. As before, the coefficients of x_i and y_i in the above constraint are the same αs. Thus, the total number of unknowns is 2 (upto scale). Each view gives two equations in terms of α. Therefore, we need one view of the three points to compute all the αs.

Lemma 2 We need one view of three points on a 3D line to obtain a linear view independent constraint on the projections of the three points.

2.1.3 Points on a plane

In $\mathbb{P}^2$ space there are three basis vectors, hence the matrices $\mathbf{C}_x$ and $\mathbf{C}_y$ constructed from a point configuration that lies in a two dimensional space, would have at most rank 3, implying that the determinant of any of their 4×4 minor would be 0. Therefore,

$$\left|\begin{bmatrix} P_{1x} & P_{1y} & 1 & x_1 \\ P_{2x} & P_{2y} & 1 & x_2 \\ P_{3x} & P_{3y} & 1 & x_3 \\ P_{4x} & P_{4y} & 1 & x_4 \end{bmatrix}\right| = \left|\begin{bmatrix} P_{1x} & P_{1y} & 1 & y_1 \\ P_{2x} & P_{2y} & 1 & y_2 \\ P_{3x} & P_{3y} & 1 & y_3 \\ P_{4x} & P_{4y} & 1 & y_4 \end{bmatrix}\right| = 0$$

which give the constraint

$$\sum_i^4 \alpha_i x_i = 0$$
$$\sum_i^4 \alpha_i y_i = 0$$

where the αs are view-independent and are the same for the constraints on x_i and y_i. The total number of unknowns is 3 upto scale and we would need two views of the four points to compute the unknown αs.

Lemma 3 We need two views of four points on a plane to obtain a linear view independent constraint on the projections of the four points.

2.1.4 Points on 3D lines

We now consider the case of a point configuration wherein the points lie on two or more 3D lines. The possible ways of arranging five points on 3D lines are

- One Line : All five points are collinear i.e. all of them lie on the same 3D line. This scenario has already been dealt in Section 2.1.2.

- Two Lines : Three points lie on one line and the rest two lie on the other line. In this case, the third point on the first line can be expressed in terms of the other two points on that line. Therefore the matrices $\mathbf{C}_x$ and $\mathbf{C}_y$ would have a maximum rank of 4, giving us the linear constraint

$$\sum_i^5 \alpha_i x_i = 0$$
$$\sum_i^5 \alpha_i y_i = 0$$

  which needs 2 views of the five points to compute the unknown αs upto scale.

- Two Lines: Four points are collinear, while one is not. In this case, we can express two points on the line passing through the four collinear points in terms of the other two points. The matrices $\mathbf{C}_x$ and $\mathbf{C}_y$ would be of rank 3, giving us the linear constraint

$$\sum_i^4 \alpha_i x_i = 0$$
$$\sum_i^4 \alpha_i y_i = 0$$

  which needs 2 views of the four points to compute the unknown αs upto scale.

- More than Two Lines : This is the same as the general configuration which has already been dealt with in Section 2.1.1.

Lemma 4 For a linear view independent constraint on the projections of points lying on two or more 3D lines, depending on the number of 3D lines involved, we need 3 to 5 points and 1 to 2 views.

2.1.5 Points on Concurrent 3D Lines

This case is different from the one of points on 3D lines described in Section 2.1.4, only if one of the points being considered is the point of intersection of the lines. There are two possible scenarios

- Two Lines : Two lines have two points while the fifth point is the point of intersection of the lines. In this case, one point on each of the lines can be expressed in terms of the other point and the point of intersection. Thus the matrices $\mathbf{C}_x$ and $\mathbf{C}_y$ would be of rank 3, giving us the constraints

$$\sum_i^4 \alpha_i x_i = 0$$
$$\sum_i^4 \alpha_i y_i = 0$$

  which need 2 views of the four points to compute the unknown αs upto scale.

- Three Lines : Two points on one line and one each on the other two with the fifth point being the point of intersection of the three lines. There are 4 independent points in this set up, the same as would be in the case of a general configuration, giving us identical constraints and requirements for their computation.

$$\sum_i^5 \alpha_i x_i = 0$$
$$\sum_i^5 \alpha_i y_i = 0$$

- More than Three Lines: This would be the same as a general configuration, with the constraint

$$\sum_i^5 \alpha_i x_i = 0$$
$$\sum_i^5 \alpha_i y_i = 0$$

Lemma 5 For a linear view independent constraint on the projections of points lying on concurrent 3D lines, one of the points being the point of intersection of the lines, we need 4 to 5 points and 2 views.

2.1.6 Points on a line on a Plane

This is the same as the case of points on a 3D line. The constraints and the requirements for their computation are identical as in Section 2.1.2.

Lemma 6 We need one view of three points on a line on a plane to obtain a linear view independent constraint on the projections of the three points.

2.1.7 Points on lines on a Plane

This is the same as the case of points on a plane, there are three independent points, giving us the same constraints as in Section 2.1.3 with identical computation requirements.

Lemma 7 We need two views of four points on lines on a plane to obtain a linear view independent constraint on the projections of the four points.

2.1.8 Points on Concurrent lines on a plane

This case is different from the case of points on lines on a plane if one of the points is the point of intersection of the lines. There are three possibilities

- Two Lines: Two points lie on each of the two lines and the fifth point lies on the point of intersection of the two lines. The second point on each line can be expressed in terms of the first line and the point of intersection. Thus the matrices $\mathbf{C}_x$ and $\mathbf{C}_y$ would be of rank 3, giving us the constraints

$$\sum_i^4 \alpha_i x_i = 0$$
$$\sum_i^4 \alpha_i y_i = 0$$

  which need 2 views of four points (two points on one line, one on the second line and the point of intersection) in to compute the unknown αs upto scale.

- More than Two Lines: This case is the same as the case of points on a plane, there are are 3 independent points, giving us the same constraints and requirements for their computation as in Section 2.1.3.

Lemma 8 We need two views of four points on concurrent lines on a plane to obtain a linear view independent constraint on the projections of the four points.

The constraints for various point configurations and their computational requirements are summarised in Table 2.1.

Configuration	Requirements for Constraint Computation
General 3D Configuration	2 Views of 5 Points
Points on a 3D Line	1 View of 3 Points
Points on a Plane	2 Views of 4 Points
Points on 3D Lines	1 to 2 Views of 3 to 5 Points
Points on Concurrent 3D Lines	2 Views of 4to 5 Points
Points on a Line on a Plane	1 View of 3 Points
Points on Lines on a Plane	2 Views of 4 Points
Points on Concurrent Lines on a Plane	2 Views of 4 Points

Table 2.1: Requirements for computation of linear view independent constraints for static points in the world.

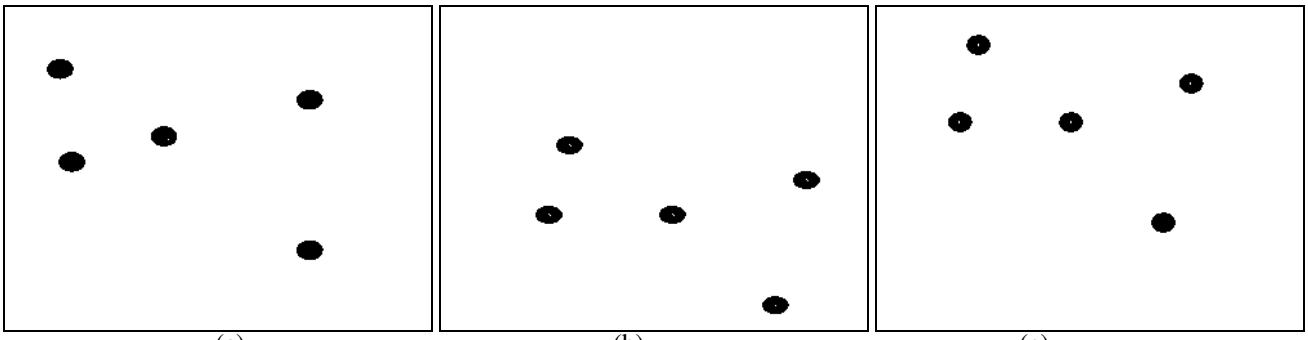

Figure 2.1: Three views of a static configuration of five points in general position

2.1.9 Experiments

A number of experiments were conducted to ascertain the validity of these constraints on simulated data. We constructed many views of various configurations. In all cases the view independent coefficients computed were found to be correct. We show results from one experiment on a general configuration of 3D points. Figure 2.1 shows three views of a static configuration of 5 points in general position. The view-independent coefficients α_is were computed from the first two views. When they were applied to the point locations in the third view, we found that the residual errors in the Equations 2.2 were almost zero.

2.2 Point Configurations in Motion

The study of non-rigid motion in the view-independent constraint framework is difficult. The case of multiple objects moving with independent uniform velocities or accelerations can be considered very close to the case of non-rigid motion. The view-independent relationships between projections of points moving with uniform velocity presented recently [36] fall under this category. The two view constraints on points moving with a constant velocity is another noteworthy contribution in this direction [35].

In this Section, we study view-independent algebraic relationships between moving objects imaged from different viewpoints. We extend the philosophy of factoring out camera parameters as done above to derive time-dependent and time-independent relationships between the velocities and accelerations of the affine projections of objects moving with independent velocities or accelerations.

2.2.1 Uniform Velocity Motion

Let $\mathbf{P}$ be a 3D world point, moving with uniform linear velocity. Let $\mathbf{I} = \begin{bmatrix} I_x & I_y & I_z \end{bmatrix}^{\mathrm{T}}$ be its initial position and $\mathbf{U} = \begin{bmatrix} U_x & U_y & U_z \end{bmatrix}^{\mathrm{T}}$ be its world velocity. Its position at any time instant t is given by

$$\mathbf{P} = \begin{bmatrix} \mathbf{I} \\ 1 \end{bmatrix} + \begin{bmatrix} \mathbf{U} \\ 0 \end{bmatrix} t \tag{2.3}$$

Let an affine camera observe the motion of the point. Let $\mathbf{p}_t^l = \begin{bmatrix} x_t^l & y_t^l & 1 \end{bmatrix}^{\mathrm{T}}$ be the projection of $\mathbf{P}$ in view l at time t due to the affine camera matrix $\mathbf{M}^l = \begin{bmatrix} \mathbf{m_1}^l & m_{14}^l \\ \mathbf{m_2}^l & m_{24}^l \\ \mathbf{0} & 1 \end{bmatrix}$. Then,

$$\mathbf{p}_t^l = \mathbf{M}^l \begin{bmatrix} \mathbf{I} \\ 1 \end{bmatrix} + \mathbf{M}^l \begin{bmatrix} \mathbf{U} \\ 0 \end{bmatrix} t. \tag{2.4}$$

Differentiating with respect to t, we get $\tilde{\mathbf{v}}^l = \mathbf{M}^l \begin{bmatrix} \mathbf{U} \\ 0 \end{bmatrix}$ where $\tilde{\mathbf{v}}^l = [v_x^l, v_y^l, 0]^{\mathrm{T}}$ is the velocity vector in the image. This implies that the velocity of a point in the image is a projection of the world velocity. Since the projection is linear (the last row of the projection matrix is [0 0 0 1]) the projection of a point moving with constant velocity in the world moves with constant velocity in the image.

The above can be expanded as

$$v_x^l = \mathbf{m_1}^l . \begin{bmatrix} U_x & U_y & U_z \end{bmatrix} \tag{2.5}$$
$$v_y^l = \mathbf{m_2}^l . \begin{bmatrix} U_x & U_y & U_z \end{bmatrix} \tag{2.6}$$

which can be written as

$$\begin{bmatrix} U_x & U_y & U_z & v_x \end{bmatrix} \begin{bmatrix} \mathbf{m_1}^l & -1 \end{bmatrix}^T = 0$$

and

$$\begin{bmatrix} U_x & U_y & U_z & v_y \end{bmatrix} \begin{bmatrix} \mathbf{m_2}^l & -1 \end{bmatrix}^T = 0$$

If we have 4 points in the scene, $P_i,\ 1 \leq i \leq 4$, with world velocities $\begin{bmatrix} U_{ix} & U_{iy} & U_{iz} \end{bmatrix}^{\mathrm{T}}$ and image velocities $\begin{bmatrix} v_{ix}^l & v_{iy}^l \end{bmatrix}^{\mathrm{T}}$, we can define a matrix $\mathbf{C}_{xv}^l$ as

$$\mathbf{C}_{xv}^l = \begin{bmatrix} U_{1x} & U_{1y} & U_{1z} & v_{1x}^l \\ U_{2x} & U_{2y} & U_{2z} & v_{2x}^l \\ U_{3x} & U_{3y} & U_{3z} & v_{3x}^l \\ U_{4x} & U_{4y} & U_{4z} & v_{4x}^l \end{bmatrix} \tag{2.7}$$

Similarly, we can define a matrix $\mathbf{C}_{yv}^l$ with the last column having the velocity vectors in y-direction. We observe that

$$\begin{aligned} \mathbf{C}_{xv}^l \begin{bmatrix} \mathbf{m_1}^l & -1 \end{bmatrix}^T &= \mathbf{0} \quad \text{and} \\ \mathbf{C}_{yv}^l \begin{bmatrix} \mathbf{m_2}^l & -1 \end{bmatrix}^T &= \mathbf{0}. \end{aligned}$$

The matrices $\mathbf{C}_{xv}^l$ and $\mathbf{C}_{yv}^l$ must be rank deficient, i.e., $|\mathbf{C}_{xv}^l| = |\mathbf{C}_{yv}^l| = 0$. By expanding the determinants we get the following functions of the image velocities

$$\begin{aligned} \alpha_0 v_{1x}^l + \alpha_1 v_{2x}^l + \alpha_2 v_{3x}^l + \alpha_3 v_{4x}^l = 0 \\ \alpha_0 v_{1y}^l + \alpha_1 v_{2y}^l + \alpha_2 v_{3y}^l + \alpha_3 v_{4y}^l = 0 \end{aligned} \tag{2.8}$$

where α's are view-independent coefficients that depend only on the world velocity parameters of the 4 points. The constraints hold for the image velocities of the projections of the four points for the same α's irrespective of the pose and intrinsic parameters of the camera. They are also time-independent as the time term has been eliminated.

Equation 2.8 has three unknowns upto scale, with each view providing two equations. Therefore, we need two views of the four points to determine all the α's up to scale.

These results are similar to the Recognition Polynomials and Shape Tensors presented or discovered earlier. It was shown that polynomials to recognize a configuration of stationary points could be constructed from 2 views of 4 points under orthographic projections [52]. This was extended to recognize human gait using 2 views of 5 points under scaled-orthographic projections [35]. Time-dependent constraints involving a single view of 5 points with uniform velocity is presented in [36] for affine projection. Our results yield view and time independent constraints involving 4 points in 2 views under general affine projection – which is a significant theoretical advancement.

2.2.2 Uniform Acceleration Motion

We now derive relationships between points when they move with constant acceleration. Let $\mathbf{P}$ be a 3D world point, moving with uniform linear acceleration. Its position at any time instant t is given by

$$\mathbf{P} = \begin{bmatrix} \mathbf{I} \\ 1 \end{bmatrix} + \begin{bmatrix} \mathbf{U} \\ 0 \end{bmatrix} t + \frac{1}{2} \begin{bmatrix} \mathbf{A} \\ 0 \end{bmatrix} t^2 \tag{2.9}$$

where $\mathbf{I}$ is the initial position of the point in inhomogeneous coordinates, $\mathbf{U}$ is its initial velocity and $\mathbf{A} = \begin{bmatrix} A_x & A_y & A_z \end{bmatrix}^{\mathrm{T}}$ is its constant acceleration.

Factoring out the camera matrices in the same way as in the previous subsection, we get the singular matrix $\mathbf{X_a}$

$$\mathbf{X_a}^l = \begin{bmatrix} (U_{1x} + A_{1x}t) & (U_{1y} + A_{1y}t) & (U_{1z} + A_{1z}t) & v^l_{1x} \\ (U_{2x} + A_{2x}t) & (U_{2y} + A_{2y}t) & (U_{2z} + A_{2z}t) & v^l_{2x} \\ (U_{3x} + A_{3x}t) & (U_{3y} + A_{3y}t) & (U_{3z} + A_{3z}t) & v^l_{3x} \\ (U_{4x} + A_{4x}t) & (U_{4y} + A_{4y}t) & (U_{4z} + A_{4z}t) & v^l_{4x} \end{bmatrix}$$

A similar singular matrix $\mathbf{Y_a}^l$ with motion parameters in y-direction also exists. Expanding $|\mathbf{X_a}^l|$ and $|\mathbf{Y_a}^l|$, we get

$$\begin{aligned} &(\beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 t^3) v^l_{1x} + \\ &(\beta_4 + \beta_5 t + \beta_6 t^2 + \beta_7 t^3) v^l_{2x} + \\ &(\beta_8 + \beta_9 t + \beta_{10} t^2 + \beta_{11} t^3) v^l_{3x} + \\ &(\beta_{12} + \beta_{13} t + \beta_{14} t^2 + \beta_{15} t^3) v^l_{4x} \quad = \quad 0 \end{aligned} \tag{2.10}$$

$$\begin{aligned} &(\beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 t^3) v^l_{1y} + \\ &(\beta_4 + \beta_5 t + \beta_6 t^2 + \beta_7 t^3) v^l_{2y} + \\ &(\beta_8 + \beta_9 t + \beta_{10} t^2 + \beta_{11} t^3) v^l_{3y} + \\ &(\beta_{12} + \beta_{13} t + \beta_{14} t^2 + \beta_{15} t^3) v^l_{4y} \quad = \quad 0 \end{aligned} \tag{2.11}$$

where the β's are view-independent coefficients that depend only on the parameters of motion of the 3D points in the world. The above relations are time-dependent and also view-independent. That is, the same β's hold no matter what the pose and intrinsic parameters of the affine camera used to view them. There are 16 unknowns ($\beta_0 \ldots \beta_{15}$) in the above relation, with each time instant providing 2 equations. We, therefore, need the velocities of 4 points at 8 time instants for computing the β's. Note that these β's can be computed from a single view, as opposed to the α's for the case of constant velocity, which needed two views. This is the direct result of time-dependence.

Type	Conditions	Time Invariant	Source
Stationary	Multiple	Yes	Many
Uniform V	5 pts, 8 frames	No	Levin et al.
Uniform V	4 pts, 2 views	Yes	This report
Uniform A	4 pts, 9 frames	No	This report
Uniform A	4 pts, 2 views	Yes	This report
Uniform ω	6 pts	No	Levin et al.

Table 2.2: Summary of the multiview constraints on a configuration of points

We now proceed to derive time-independent constraints for the case of constant linear acceleration in the world. For this, we differentiate the constant acceleration motion equation (Equation. 2.9) twice to get

$$a_x^l = \mathbf{m_1}^l . \begin{bmatrix} A_x & A_y & A_z \end{bmatrix} \tag{2.12}$$

$$a_y^l = \mathbf{m_2}^l . \begin{bmatrix} A_x & A_y & A_z \end{bmatrix} \tag{2.13}$$

where a_x^l and a_y^l are the image-accelerations of the projection of the point. We can now define singular matrices $\mathbf{X'_a}^l$ and $\mathbf{Y'_a}^l$ for the four points P_i, $1 \leq i \leq 4$ as in the case of uniform velocity. $\mathbf{X'_a}^l$ is given below.

$$\mathbf{X'_a}^l = \begin{bmatrix} A_{1x} & A_{1y} & A_{1z} & a_{1x}^l \\ A_{2x} & A_{2y} & A_{2z} & a_{2x}^l \\ A_{3x} & A_{3y} & A_{3z} & a_{3x}^l \\ A_{4x} & A_{4y} & A_{4z} & a_{4x}^l \end{bmatrix} \tag{2.14}$$

Expanding the determinants of $\mathbf{X'_a}^l$ and $\mathbf{Y'_a}^l$, we get

$$\begin{aligned} \gamma_0 a_{1x}^l + \gamma_1 a_{2x}^l + \gamma_2 a_{3x}^l + \gamma_3 a_{4x}^l = 0 \\ \gamma_0 a_{1y}^l + \gamma_1 a_{2y}^l + \gamma_2 a_{3y}^l + \gamma_3 a_{4y}^l = 0, \end{aligned} \tag{2.15}$$

where the γ's are functions of world accelerations parameters of the 4 points only. The γ's are view and time independent. The system of Equations 2.15 has three unknowns (upto scale) and since each view gives us two equations we need 2 views of the 4 points to determine all the γ's (upto scale).

Levin et al. [36] derive constraints for motion constrained to elliptic paths. It is the first time that view or time independent constraints for points moving with constant linear acceleration have been derived.

A summary of the view independent constraints on a configuration of stationary points or a configuration of points undergoing structured motion is presented in Table 2.2.

Some preliminary results of this work were presented in [53].

2.2.3 Experiments

We conducted a number of experiments on simulated data to determine the validity of the constraints on moving points that we have derived above and ascertain their applicability to solving problems like recognition of configurations in motion. Recognition of motion using single view constraints was reported earlier [36]. Single view constraints can be used to index into motions of a person performing a sitting movement, for instance. The same philosophy can be

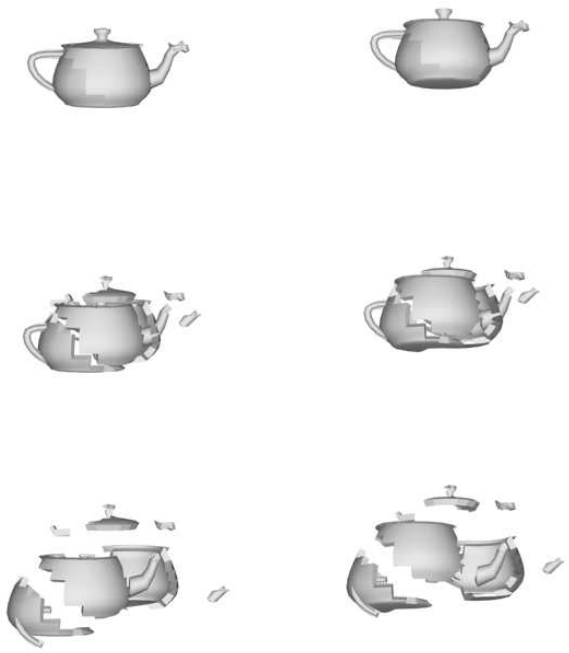

Figure 2.2: Two image sequences of an exploding pot

used for recognition using our constraints, which need less points. We considered three views of many configurations of points moving with independent velocities. From two views we determined the view-independent coefficients αs. When we applied them to the velocities of the projections in the third view we found the residual errors in Equations 2.8 to be almost zero indicating recognition of the point configuration. The residuals were non-zero when αs computed from two views of one configuration were applied to the velocities of projections of a different configuration. For configurations of points having independent accelerations we can have a similar strategy which can be extended as below if we have more than 4 points. Suppose we can track N sets of 4 points on the deforming body in all the views. From velocities of the projections at 8 time instants in each view and each set of points, a set of β's can be computed. The β's for a set of points computed from any view will satisfy the constraints expressed in Equations 2.10 and 2.11 for that set of points, in all views. This should be true for all N sets of points, for the body to be the same in all the views. To verify this strategy for recognition we considered views of a simulated explosion of a teapot, with the various fragments flying off in different directions with independent constant accelerations. Figure 2.2 shows a few frames from the image sequences of the exploding tea pot from two different view points. A sets of points on the tea pot were tracked across the explosion in all the views. It was found that the β values computed for each set of points from one view were valid for the same set of points in all views. The coefficients computed were highly similar for a specific motion and different for different motions. These coefficients may be used for gait recognition or similar applications.

These constraints can also be used for applications like alignment of frames of synchronized videos which is discussed in detail in Chapter 6.

Closure

In this chapter, we have used an approach to derive view independent constraints on views of static point configurations wherein we factored out the camera parameters. We then extended the philosophy to derive time-dependent and time-independent view independent constraints on configurations of points moving with different but uniform linear velocities or accelerations – scenarios which are very similar to that of non-rigid motion. We have also presented experimental results that assert the validity of these constraints.

We have completely defined the structure of the motion of the projections of a configuration of points and its parameters when the points move with different but uniform linear motion parameters. This attempt at modeling non-rigid motion is valid in many cases, but one would desire to arrive at constraints that hold for general non-rigid motion. We make the observation that when a point moves in the world, its projections in various views trace out contours which can be thought to be projections of the trajectory traced out in the world by the moving point. The problem of analysing the 'arbitrary' motion of a point in the world can be mapped to the problem of analysing the 'shape' of the trajectories of the projections in the various views. Such an approach to analysing points in motion by modeling their trajectories as contours is presented in the next chapter.

Chapter 3

Modelling Trajectory as a Contour

Assumptions of linear motion with constant velocity or acceleration are valid in many situations. Can we arrive at such constraints for general non-rigid motion of points in space? That would be most beneficial. In this Chapter, we derive algebraic constraints for such a situation.

Our approach is based on the following observation. A moving 3D point traces out a *closed* contour or curve in space over time. This contour gets mapped to a contour in the image. If we have multiple views of the same object motion, their consistency reduces to the matching of corresponding shapes. A contour-matching approach can be used for this step. If the 3D motion of the point is restricted to a plane, the problem reduces to the problem of analysing planar contours. Recognition of motion becomes *planar contour recognition*. In this chapter we review planar contour recognition and their application to motion.

Many planar object recognition efforts have been reported for the simple case of similarity transformation between views [41, 54, 55, 40]. A planar object can be recognised by comparing it with the set of *a priori* known shapes. Recognition by alignment was attempted by Huttenlocher and Ullman [16]. They computed a match by determining the existence of a transformation that when applied to a model would result in the given view. Comparison can also be carried out by generating a geometric model of the boundary, as is done in algorithms based on polygonal approximation [40]. Linear or other parametric approximations of the boundary can also be used. Algorithms based on computing geometrically invariant features from the discrete set of boundary points have also been developed [4]. These features can be curvatures, compactness, moments, etc. Another class of algorithms integrate the advantages of both by modeling the boundary in a transform domain like the Fourier one as was done by Zahn and Roskies [41].

In these algorithms, the reference and test images are related by similarity transformations, involving in-plane rotations, translations and scaling. However typical shape recognition problems involve more complex transformations between views and conventional approaches based on Euclidean and similarity frameworks would be insufficient. There exists a notably different approach for recognition across multiple views. These class of algorithms consider recognition as establishing one-to-one relationships between shapes, in the presence of unknown image-to-image transformations. Ullman and Basri [18] formulated mechanisms for recognition of objects using linear combination of models for orthographic views. This result hints that the various views of an object lie in a lower dimensional linear subspace. The performance of these algorithms depend on the accuracy of the feature-to-feature correspondences. Arbter *et al.* [42] formulated techniques for affine invariant recognition in the Fourier Domain. Their emphasis was on choosing a suitable set of affine invariant features and performing matching using those features.

In an earlier work [43, 44, 45, 46, 47], we had formulated techniques for recognizing projections of planar contours in the Fourier domain when the transformation between views is affine. We present a brief overview of those techniques in this chapter and argue that these techniques can be used for recognizing the contours of the projections of

moving points. Many surveillance applications involve studying the motion of an object (like vehicles on the ground) from cameras that are far away from them (on top of tall buildings or on satellites). The trajectory of the objects is restricted to a plane and the cameras are affine in practice in this case.

Organisation of the Chapter In Section 1, we consider the simple case of points moving in straight lines with no restrictions on their velocity, acceleration, etc. The more challenging and interesting problem of arbitrary non-rigid motion is presented in Section 2. Various shape recognition techniques for recognizing the contour traced out by the projections in the various views are presented along with results of experiments conducted to verify their validity.

3.1 Linear Motion

Linear motion includes motion along a world straight line with no restrictions on the velocity, acceleration, etc. In this case, the trajectories of the points will be straight lines. Constraints that hold for matching lines in multiple views can be used on each moving point independently. For example, the Trilinear Tensor can relate lines in three views. If a world line is imaged as l^1, l^2, and l^3 in three views, they are related by a trilinear constraint [11, 56, 10].

$$l^1 = (l^2)^{\mathrm{T}} \mathcal{T} l^3 \tag{3.1}$$

where $\mathcal{T}$ is a suitable tensor.

3.2 Arbitrary Motion

Let $\mathbf{O}$ be a set of N points on the planar trajectory of a point and let $(u^l[i], v^l[i])$ be its images in view l using an affine camera. We represent this contour using a sequence of vectors of complex numbers as given below.

$$\mathbf{x}^l[i] = \begin{bmatrix} u^l[i] + j0 \\ v^l[i] + j0 \end{bmatrix}$$

Under affine projection, the image-to-image homography between a pair of views of a plane is affine also. Thus, the corresponding points of the contour in view l are related to the points in the reference view 0 by the relation (in inhomogeneous coordinates)

$$\mathbf{x}^l[i] = \mathbf{A}^l \mathbf{x}^0[i] + \mathbf{b}^l, \;\; 0 \leq i < N \tag{3.2}$$

where $\mathbf{A}^l$ is the upper 2×2 minor of the homography and $\mathbf{b}^l$ is the vector of the upper 2 elements of the third column of the homography. Taking the Fourier transform of Equation 3.2, we get

$$\mathbf{X}^l[k] = \mathbf{A}^l \mathbf{X}^0[k] + \mathbf{b}^l \delta[0], \;\; 0 \leq k < N$$

where $\mathbf{X}^l = \begin{bmatrix} U^l, & V^l \end{bmatrix}^{\mathrm{T}}$; U^l and V^l are Fourier transform sequences of u^l and v^l respectively. We call the $\mathbf{X}^l[k]$ vector Fourier coefficients. The above equation can be rewritten as (ignoring the spatial frequency corresponding to k = 0)

$$\mathbf{X}^l[k] = \mathbf{A}^l \mathbf{X}^0[k], \;\; 0 < k < N \tag{3.3}$$

In the event that we do not have alignment information across views, then Equation 3.2 becomes

$$\mathbf{x}^l[i] = \mathbf{A}^l \mathbf{x}^0[i + \lambda_l] + \mathbf{b}^l, \;\; 0 \leq i < N \tag{3.4}$$

where λ_l is the time alignment parameter, i.e. shifting the contour representation by λ_l would align corresponding points across views. Making use of the Fourier Shift Theorem Equation 3.3 now becomes

$$\mathbf{X}^l[k] = \mathbf{A}^l \mathbf{X}^0[k] e^{j2\pi\lambda_l k/N}, \;\; 0 < k < N \tag{3.5}$$

Armed with this notation that has the distinct advantage of keeping the $u - v$ coordinates separate (as opposed to some other contour representations) we explore the contour recognition techniques presented next.

Invariant Sequence We define the measure κ for the points on the contour in the view l as

$$\kappa^l[k] = (\mathbf{X}^l[k])^{*\mathrm{T}} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \mathbf{X}^l[k], \;\; 0 < k < N \tag{3.6}$$

The * denotes complex conjugate. It can be shown that [44]

$$\kappa^l[k] = |\mathbf{A}^l| \; \kappa^0[k], \;\; 0 < k < N. \tag{3.7}$$

where, $|\mathbf{A}^l|$ denotes the determinant of $\mathbf{A}^l$. The κ values defined by Equation 3.6, which can be computed independently for each view from the Fourier transform of the contour points, identify the contour formed by the motion as the κ sequence is invariant upto scale. Consider the following $M \times (N-1)$ matrix for M views of the planar contour

$$\Theta = \begin{bmatrix} \kappa^0[1] & \kappa^0[2] & \cdots & \kappa^0[N-1] \\ \kappa^1[1] & \kappa^1[2] & \cdots & \kappa^1[N-1] \\ \cdots & \cdots & \cdots & \cdots \\ \kappa^{M-1}[1] & \cdots & \cdots & \kappa^{M-1}[N-1] \end{bmatrix}. \tag{3.8}$$

It can be seen from Equation 3.7 that

$$rank(\Theta) = 1.$$

(See reference [44] for a detailed discussion on rank constraints for shape matching.)

This constraint is view independent as the κ measure is computed independently in each view. There are no restrictions on the number of views or frames per se. In practice, however, the Fourier transform will be reliable only if the curve has sufficient length. The motion of the point is arbitrary. If a number of points can be tracked independently, each contour will yield a different constraint, all of which have to be satisfied simultaneously. It is clear that non-rigid motion is also covered by these constraints.

The above result hints that there can exist a number of algebraic constraints on the trajectory traced out by the projections of a moving point in a view. We now present two more such constraints.

Phase based constraints The κ measure correlates each Fourier coffined with itself. What happens when we correlate each Fourier coffined with a fixed one? We modify the definition of κ as follows

$$\kappa'_p(l)[k] = (\mathbf{X}^l[k])^{*\mathrm{T}} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \mathbf{X}^l[p], \quad 0 < k < N$$

for any fixed $p \neq 0$. It can be shown [44] that

$$\kappa'_p(l)[k] = |\mathbf{A}^l| \; \kappa'_p(0)[k] \; e^{-j2\pi\lambda_l(k-p)/N} \tag{3.9}$$

Equation 3.9 states that the phases of $\kappa'_p(l)$ and $\kappa'_p(0)$ differ by an amount proportional to the shift λ_l and the differential frequency $k-p$. Therefore, the ratio $\frac{\kappa'_p(l)}{\kappa'_p(0)}$ will be a complex sinusoid $ce^{-j2\pi\lambda_l(k-p)/N}$. The value of λ_l can be computed from the inverse Fourier transform of the quotient series. Thus, the phases of $\kappa'_p(l)$ can be used as a signature for the contour. We can form a $M \times (N-1)$ matrix Θ', similar to the one above, that stacks the phases of $\kappa'_1(l)$ (taking $p=1$). It will have the form $\Theta' =$

$$\begin{bmatrix} \theta_1 & \theta_2 & \theta_3 & \dots & \theta_{N-1} \\ \theta_1 & \theta_2+\phi_1 & \theta_3+2\phi_1 & \dots & \theta_{N-1}+(N-2)\phi_1 \\ \dots & \dots & \dots & \dots & \dots \\ \theta_1 & \theta_2+\phi_{M-1} & \theta_3+2\phi_{M-1} & \dots & \theta_{N-1}+(N-2)\phi_{M-1} \end{bmatrix} \tag{3.10}$$

where θ_i are the phases of $\kappa'_1(0)$ and $\phi_l = -2\pi\lambda_l/N$. This matrix will have a rank of 2 irrespective of M. Therefore, the rank constraint on the above matrix, which is a necessary condition for the contours in two affine transformed views to be corresponding is

$$rank(\Theta') = 2.$$

We see that κ' can be computed from a single view. Thus, the phases of κ' values provide a truly view-independent description of the trajectory of the projection of the moving point.

Magnitude based Constraints Unless properly taken care, the phase based algebraic constraints can have problems with the phase wrap around. We now present a rank-three constraint based on magnitudes of the vector Fourier coefficients. Let $\mathbf{U}^l[k] = \mathbf{U}^l_R[k] + j\mathbf{U}^l_I[k]_I$ and $\mathbf{V}^l[k] = \mathbf{V}^l_R[k] + j\mathbf{V}^l_I[k]$ where the subscripts R and I denote real and imaginary parts of the respective complex number. It can be shown [45] that

$$\begin{aligned} |U^l[k]|^2 &= (a^l_{11})^2[(U^0_R[k])^2 + (U^0_I[k])^2] + \\ &\quad (a^l_{12})^2[(V^0_R[k])^2 + (V^0_I[k])^2] + \\ &\quad 2a^l_{11}a^l_{12}[U^0_R[k]V^0_R[k] + U^0_I[k]V^0_I[k]] \\ |V^l[k]|^2 &= (a^l_{21})^2[(U^0_R[k])^2 + (U^0_I[k])^2] + \\ &\quad (a^l_{22})^2[(V^0_R[k])^2 + (V^0_I[k])^2] + \\ &\quad 2a^l_{21}a^l_{22}[U^0_R[k]V^0_R[k] + U^0_I[k]V^0_I[k]] \end{aligned} \tag{3.11}$$

where $\mathbf{A}^l = a^l_{ij}$. Its evident from Equation 3.11 that the magnitude of the components of the Fourier domain representation in any view can be expressed in terms of the components in a reference view. This result can be expressed in the following manner. Given M views, we can construct a $(2M+1) \times (N-1)$ matrix as follows. The first row consists of the sum of products $(U^0_R[k]V^0_R[k] + U^0_I[k]V^0_I[k])$, 0 being the reference view. Every view contributes two rows to this matrix (except the reference view, which contributes 3 rows) the magnitudes of U in one row and the

magnitudes of V in the other. Let $\Theta'' =$

$$\begin{bmatrix} (U_R^0[1]V_R^0[1] + U_I^0[1]V_I^0[1])) & ... & (U_R^0[G]V_R^0[G] + U_I^0[G]V_I^0[G]) \\ ((U_R^0[1])^2 + (U_I^0[1])^2) & ... & ((U_R^0[G])^2 + (U_I^0[G])^2) \\ ((V_R^0[1])^2 + (V_I^0[1])^2) & ... & ((V_R^0[G])^2 + (V_I^0[G])^2) \\ ((U_R^1[1])^2 + (U_I^1[1])^2) & ... & ((U_R^1[G])^2 + (U_I^1[G])^2) \\ ((V_R^1[1])^2 + (V_I^1[1])^2) & ... & ((V_R^1[G])^2 + (V_I^1[G])^2) \\ ((U_R^2[1])^2 + (U_I^2[1])^2) & ... & ((U_R^2[G])^2 + (U_I^2[G])^2) \\ ((V_R^2[1])^2 + (V_I^2[1])^2) & ... & ((V_R^2[G])^2 + (V_I^2[G])^2) \\ ... & ... & ... \\ ((U_R^M[1])^2 + (U_I^M[1])^2) & ... & ((U_R^M[G])^2 + (U_I^M[G])^2) \\ ((V_R^M[1])^2 + (V_I^M[1])^2) & ... & ((V_R^M[G])^2 + (V_I^M[G])^2) \end{bmatrix} \tag{3.12}$$

(using G for $(N-1)$) From Equation 3.11 one can conclude that the rank of Θ'' is 3, irrespective of the number of views. Therefore, the constraint,

$$rank(\Theta'') = 3 \tag{3.13}$$

is a necessary constraint on the projection of the trajectory in the various views.

3.2.1 Experiments

We conducted a number of experiments on simulated data to validate the constraints presented above. The trajectories of the projections of a point having planar motion were recorded in the various views and used to construct corresponding contours. The above contour recognition strategies were then evaluated and found them to be valid. The rank constraint on Θ was found to be very robust, while the other two constraints were sensitive to noise. The rank of matrices were determined using Singular Value Decomposition (SVD) [57]. The number of non-zero singular values of a matrix gives the rank of the matrix. When the contour is represented using integer coordinates, discretization introduces errors that make the rank constraint an approximation, but nonetheless clearly enforceable. To verify whether a matrix has an approximate rank r, we consider the ratio of rth to $(r+1)$th singular values of the matrix. This ratio is high if the matrix has an approximate rank of r. In our experiments, if the rth singular value was found to be greater than the $(r+1)$th singular value by more than an order of 2, we concluded that the rank of the matrix was r.

Closure

Graduating from analysing projections of structured motion in the world (Chapter 2), we have analysed the trajectories of points having an 'unstructured' world motion model in this chapter. We made the observation that when a point moves in the world it traces out a contour in the world and the contour traced out in the image by the projection of the world point over time is the projection of the world contour. Analysing motion is thus equivalent to analysing the contour traced by the projection. If the motion is restricted to a line then the Trilinear Tensor can be used to relate the trajectories. If the point's motion in the world is arbitrary but restricted to a plane then the problem becomes one of analysis of projections of a planar contour, solutions to which have been presented here.

In chapter 2, we have looked at the structure of the motion of the projection of a point that moves in the world following a uniform motion model. In this chapter, we have seen a number of shape constraints that we have used to recognize trajectories of points. The question that springs up is if we have shape constraints and structured motion constraints then can we combine the two so as to recognize a shape that is deforming in a structured manner? This question is answered in the next chapter.

Chapter 4

Recognizing Deforming Contours

In Chapter 2 we had examined the structure of the motion of the projection of a point moving with uniform linear velocity or acceleration in the world. In Chapter 3, we had looked at a number of shape recognition techniques that can be used for analysing non-rigid motion by considering the shape or contour that is traced out by the projection of the moving point. The question then arises as to whether we can combine the motion and shape constraints to arrive at a mechanism to recognize a contour undergoing structured deformation which we define as a contour whose points move with independent uniform linear velocity or acceleration. We specifically consider the case when a planar contour deforms in a manner such that it always lies in the same plane.

Recognition of deformable shapes has been studied and applied to tracking of non-rigid objects [58, 59] when the deformation between two consecutive frames is small, handwriting recognition [60, 61], and contour extraction and modeling [62, 63]. Some approaches suggest learning a deformable model from examples, while some use deformable templates and ascertain a match by determining how much a template has to be deformed to get the test shape. These techniques do not assume any specific structure in the deformation. Our work on the other hand attempts to achieve a sound theoretical recognition mechanism when the deformation has a particular structure.

Organisation of the chapter The shape and motion constraints that we would be using are outlined in Section 1. In Section 2 we outline a recognition technique for case when we are able to track points only in the reference view while a mechanism for dealing with the situation when we are able to track points in all views is presented in Section 3. Section 4 presents a few comments on experimental evaluation of these constraints.

4.1 Deforming Contour with Coplanar Velocities

Let $\mathbf{P}[i]$ be N points on a plane with different but coplanar uniform velocities $\mathbf{V}[i]$. Let the projection of $\mathbf{P}[i]$ in view l at time t be $\mathbf{p}_t^l[i]$. Since the points lie on the same plane all through out there exists a homography that maps points in one view to points in the other. When the cameras imaging the scene are affine, the views of a plane are related by an affine homography. in which case the image-to-image transformation between the views for corresponding time instants becomes

$$\mathbf{p}_t^l[i] = \mathbf{A}^l \mathbf{p}_t^0[i] + \mathbf{b}^l, \quad 0 \leq i < N \tag{4.1}$$

where $\mathbf{p}_t^l[i]$ is the inhomogeneous representation of the ith point in view l at time t and $\mathbf{A}^l$ and $\mathbf{b}^l$ are as in Equation 3.2.

Taking the component-wise Fourier Transform we get,

$$\bar{\mathbf{P}}_t^l[k] = \mathbf{A}^l \bar{\mathbf{P}}_t^0[k] + \mathbf{b}^l \ \delta(0), \quad 0 \leq k < N$$

where $\bar{\mathbf{P}}_t^l$ is the Fourier Transform of $\mathbf{p}_t^l$. We can eliminate the $\mathbf{b}^l$ term by dropping the DC term ($k = 0$) to get

$$\bar{\mathbf{P}}_t^l[k] = \mathbf{A}^l \bar{\mathbf{P}}_t^0[k], \quad 0 < k < N \tag{4.2}$$

The above expressions are valid for the scenarios when correspondence between points across views is known. In case correspondence information across views is not available Equations 4.1 and 4.2 assume the form

$$\begin{aligned} \mathbf{p}_t^l[i] &= \mathbf{A}^l \mathbf{p}_t^0[i + \lambda_l] + \mathbf{b}^l, \quad 0 \leq i < N \\ \bar{\mathbf{P}}_t^l[k] &= \mathbf{A}^l \bar{\mathbf{P}}_t^0[k] e^{j\omega k \lambda_l}, \ \ 0 < k < N, \omega = \frac{2\pi}{N} \end{aligned} \tag{4.3}$$

where λ_l is the unknown shift that aligns the corresponding points of $\mathbf{p}_t^0$ and $\mathbf{p}_t^l$.

In this way we model the shape in the Fourier domain. Next we look at the motion constraints that we would be using.

In Section 2.2.1 we showed that the velocity of a point in the image is a projection of the world velocity. Since affine projection is linear, the projection of a point moving with constant velocity in the world moves with constant velocity in the image. Therefore, the projection at any time t is given by

$$\mathbf{p}_t^l = \mathbf{p}_0^l + \mathbf{v}^l t \tag{4.4}$$

where $\mathbf{v} = \begin{bmatrix} v_x^l \\ v_y^l \end{bmatrix}$ is the velocity vector in the image. For a configuration of N points we can write Equation 4.4 as

$$\mathbf{p}_t^l[i] = \mathbf{p}_0^l[i] + \mathbf{v}^l[i] t \quad 0 \leq i < N \tag{4.5}$$

Taking the Fourier Transform of Equation 4.5 we get

$$\bar{\mathbf{P}}_t^l[k] = \bar{\mathbf{P}}_0^l[k] + \bar{\mathbf{V}}^l[k] t \tag{4.6}$$

where $\bar{\mathbf{V}}^l$ is the Fourier Transform of the sequence $\mathbf{v}^l$.

We next combine the Fourier domain representation of the motion and shape constraints to design two techniques to recognize a contour deforming in a manner such that all points on it move with independent uniform coplanar velocities.

4.2 Recognition Constraint I

In Section 3.2 we had introduced the κ measure as a means to recognize a contour. We compute the same measure on the Fourier domain representation of the deforming contour.

$$\begin{aligned} \kappa_t^l[k] &= \bar{\mathbf{P}}_t^l[k]^{*T} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \bar{\mathbf{P}}_t^l[k]. \quad 0 < k < N \\ &= (\mathbf{A}^l \bar{\mathbf{P}}_t^0[k] e^{j\omega k \lambda_l})^{*T} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} (\mathbf{A}^l \bar{\mathbf{P}}_t^0[k] e^{j\omega k \lambda_l}) \ \text{(Using Equation 4.3)} \end{aligned}$$

$$
\begin{aligned}
&= (\bar{\mathbf{P}}_t^0[k])^{*T}(\mathbf{A}^l)^T \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \mathbf{A}^l \bar{\mathbf{P}}_t^0[k] \quad , (\mathbf{A}^* = \mathbf{A}) \\
&= det(\mathbf{A}^l)(\bar{\mathbf{P}}_t^0[k])^{*T} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \bar{\mathbf{P}}_t^0[k] \\
&= det(\mathbf{A}^l)(\bar{\mathbf{P}}_0^0[k] + \bar{\mathbf{V}}^0[k]t)^{*T} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} (\bar{\mathbf{P}}_0^0[k] + \bar{\mathbf{V}}^0[k]t) \text{ (Using Equation 4.6)} \\
&= det(\mathbf{A}^l)(\bar{\mathbf{P}}_0^0[k]^{*T} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \bar{\mathbf{P}}_0^0[k] + t\bar{\mathbf{V}}^0[k]^{*T} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \bar{\mathbf{P}}_0^0[k] \\
&+ \bar{\mathbf{P}}_0^0[k]^{*T} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \bar{\mathbf{V}}^0[k]t + t\bar{\mathbf{V}}^0[k]^{*T} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \bar{\mathbf{V}}^0[k]t) \\
&= det(\mathbf{A}^l)(\alpha_1[k] + \alpha_2[k]t + \alpha_3[k]t^2) \qquad (4.7)
\end{aligned}
$$

where $det(\mathbf{A}^l)$ is the determinant of $\mathbf{A}^l$. The αs are functions of measurements made only in the reference image. They are pure imaginary quantities and can be computed by observing 2 frames in the reference view to determine $\mathbf{p}_0^0$ (before deformation starts) and $\mathbf{v}^0$ (and consequently $\bar{\mathbf{P}}_0^0$ and $\bar{\mathbf{V}}^0$). It is assumed that we are able to perform tracking in the reference view thereby enabling us to compute $\mathbf{v}^0$. However, no time synchronization or correspondence is required between view l and the reference view.

For the reference view, $\mathbf{A}^l$ is the 2×2 identity matrix. Therefore, from Equation 4.7

$$
\begin{aligned}
\kappa_t^0[k] &= \alpha_1[k] + \alpha_2[k]t + \alpha_3[k]t^2 \\
\kappa_t^l[k] &= det(\mathbf{A}^l)\kappa_t^0[k] \qquad (4.8)
\end{aligned}
$$

This is consistent with the result for discrete contours established in Section 3.2, i.e. the κ measure for multiple views of the deforming contour at the same time are multiples of each other.

Recognition

We now detail how this constraint can be used for recognition. The case of recognizing the contour when given different views of the contour at the same time instant maps to the problem of recognizing two corresponding contours and is relatively simple to solve. It is the problem of recognizing the contour when we have its views at different time instants that is more challenging.

Configuration at the same time in multiple views

This is akin to the case of recognizing a discrete contour in multiple views. Equation 4.8 provides a recognition constraint for such a case. Given M views, we can compute a $M \times (N-1)$ measurement matrix $\mathbf{C}_1$ constructed by stacking the κ_t^l measures for the various views, one row for each view. Since the various rows are scaled versions of each other, the rank of $\mathbf{C}_1$ would be 1. Therefore the recognition constraint is

$$rank(\mathbf{C}_1) = 1$$

Note that this is only a necessary constraint for recognition.

Configuration at different times in multiple views

This is a more challenging and as yet unattempted problem. Let us assume that in the reference view (0), we are able to track the points in two frames (identify points in a view across time) and hence able to identify all αs. Now given

the configuration observed in any other view at any time t, we can recognize it to be the same as the one observed in the reference view. Observe that Equation 4.7 states that κ_t^l is a linear combination of the vectors α_i, the time t being component of the linear combination coefficients. Given M views, we can construct a $(M+2) \times (N-1)$ measurement matrix $\mathbf{C}_2$ whose first three rows contain the vectors α_i, $i = 1, 2, 3$. The κ_t^l computed in the various views (except the reference view) then contribute one row each to $\mathbf{C}_2$.

$$\mathbf{C}_2 = \begin{bmatrix} \alpha_1[1] & \alpha_1[2] & \dots & \alpha_1[N-1] \\ \alpha_2[1] & \alpha_2[2] & \dots & \alpha_2[N-1] \\ \alpha_3[1] & \alpha_3[2] & \dots & \alpha_3[N-1] \\ \kappa_{t1}^1[1] & \kappa_{t1}^1[2] & \dots & \kappa_{t1}^1[N-1] \\ \kappa_{t2}^2[1] & \kappa_{t2}^2[2] & \dots & \kappa_{t2}^2[N-1] \\ \dots & \dots & \dots & \dots \\ \kappa_{t(m-1)}^{(M-1)}[1] & \kappa_{t(m-1)}^{(M-1)}[2] & \dots & \kappa_{t(m-1)}^{(M-1)}[N-1] \end{bmatrix}$$

Note the time instants at which κ is computed in a view need not be the same in all views. Since, every row constructed from κ_t^l can be expressed as a linear combination of the first 3 rows, the recognition constraint is

$$rank(\mathbf{C}_2) = 3$$

Note that this technique does not need correspondence across views and assumes tracking only in the reference view. This technique also does not depend on the time instant at which the κ values are computed in a view. In fact, this technique can be used to determine the time parameter and hence can be used to achieve alignment of frames of synchronized videos which is dealt in detail in Chapter 6.

4.3 Recognition Constraint II

We now consider the case wherein we are able to perform tracking in all views for atleast two frames. If we are able to do so, then we can determine the the distance (velocity scaled by time) of the projection from its initial position in each view

$$\mathbf{v}^l[i] * t = \mathbf{p}_t^l[i] - \mathbf{p}_0^l[i]$$

The image velocities are related by the same transformation as corresponding points. Therefore,

$$\mathbf{v}^l[i] = \mathbf{A}^l \mathbf{v}^0[i] + \mathbf{b}^l, \quad 0 \leq i < N$$

Computing the κ measure on the Fourier domain representation of the scaled velocity, obtained by subtracting the position of the projections at two time instants, we get

$$\kappa^l[k] = t^2 * det(\mathbf{A}^l)\kappa^0[k] \tag{4.9}$$

(Assuming that two successive time instants were chosen in the reference view 0.) This constraint can be used for recognition. Given M views, we construct a $M \times (N-1)$ measurement matrix $\mathbf{C}_3$, the κ values computed from the Fourier domain representation of the time scaled velocities in each view contributing one row each. Since all the κs in the various views are scaled versions of each other, the necessary constraint for recognition becomes

$$rank(\mathbf{C}_3) = 1$$

This technique only assumes we are able to track points in each view. It does not need correspondence across views. It is can be used to also determine the time difference between two views of a configuration and hence can also be used to solve the problem of aligning frames of synchronized videos.

4.4 Experiments

We conducted experiments on simulated data to verify the validity of the constraints for recognizing contours deforming in a structured manner. The constraints were found to be valid in most of the cases. These constraints are however very sensitive to noise and so failed in some experiments.

Closure

In this chapter, we have brought together shape and motion constraints to derive recognition mechanisms for contours that deform in a structured manner – the points on the contour move with independent uniform linear velocity. As with other constraints presented in this report, we can design analogous techniques for the case of points on the contour moving with independent uniform linear accelerations.

The constraints derived and the algorithms designed till now have been for the case of affine cameras. Affine cameras model many scenarios in multiview imaging and so these results are significant. However, there are many situations wherein affine cameras fail to approximate general projective cameras and so it is desirable to derive constraints and design algorithms for the case of general projective cameras. This objective is pursued in the next chapter.

Chapter 5

Constraints on Point Configurations for Projective Cameras

We have determined a number of constraints on the affine projections of a configuration of points when the points are stationary and when the points move with independent linear motion parameters. To accommodate for general non-rigid motion, we modeled the trajectory of the point as a contour and transformed the problem of motion analysis to that of shape analysis. We then combined shape and motion constraints to develop a novel recognition mechanism to recognize a contour that deforms such that all points on its boundary move with independent uniform linear velocity or acceleration. All these have been designed to work when the cameras used are affine. Though an affine camera assumption seems to be valid in many cases in multiview imaging, it is desirable to arrive at constraints derived assuming a projective camera.

Levin et al. [36] have derived view-independent constraints on the projections of points moving with constant velocity. For the case when the points of the configuration of interest are in general position, their constraint has 90 unknowns and is of order 9 in the time parameter and requires 6 points and 49 time instants for their computation. This constraint is too complex to be used, which the authors also acknowledge. They then proceed to derive constraints for more specific configurations like configurations having coplanar trajectories. Their work is the closest to ours. In this chapter, we first try to extend the philosophy of factoring out camera parameters to arrive at view-independent constraints for projective camera projections of point configurations that have independent linear motion models in the world. After that we develop a novel parameterization for the projections of points moving as per a uniform linear motion model in the world and use that to derive view-independent constraints on a configuration of such points.

5.1 View-Independence by Factoring Out the Camera

In chapter 2, we have derived view-independent constraints by factoring out the camera parameters using a suitable number of points. We try to extend that philosophy to the deriving a similar constraint. We consider the case of uniform linear velocity. Suppose that we have a world point P that moves with uniform linear motion as per Equation 2.3. Let this point be viewed by a perspective camera represented by the camera matrix $\mathbf{M}$. The projection of P will trace a line in the image. Let us represent this line l by $\begin{bmatrix} a & b & c \end{bmatrix}^{\mathrm{T}}$. Let p be the projection of P due to $\mathbf{M}$ at some time instant t. Since p will lie on the line of motion l we have

$$l^T p = 0$$

Writing p as $\mathbf{MP}$ we get

$$\begin{bmatrix} a & b & c \end{bmatrix} \mathbf{M} \begin{bmatrix} (I_x + U_x t) & (I_y + U_y t) & (I_z + U_z t) & 1 \end{bmatrix}^T = 0$$

Differentiating w.r.t. t, gives us

$$(a\mathbf{m_1} + b\mathbf{m_2} + c\mathbf{m_3}).\begin{bmatrix} U_x & U_y & U_z \end{bmatrix} = 0$$

where $\mathbf{m_i}$ is the vector of the first three elements in the ith row of $\mathbf{M}$. Proceeding in a similar manner as before to eliminate the projection terms m_{ij}, we can derive a view-independent relation for uniform velocity motion under perspective projection. However, the number of unknowns are very high and hence the number of views or time instants needed to solve for them runs into large numbers. Extension to constant acceleration motion also has similar problems under perspective projection. Thus, this approach provides a very complex view-independent constraint. Considering the case of coplanar motion - all the points of the configuration remain on the same plane for the entire duration, also does not reduce the complexity of the constraint to reasonable levels.

In the next section, we explore an alternate approach to deriving view independent constraints for coplanar motion. We first parameterize the location of the projection of a point in a view and then compute an invariant for the configuration of points (cross ratio of areas, etc). Since the parameterization has time as one of the parameters, the invariant will also be time varying. Therefore such a constraint would be time-dependent and view-independent.

5.2 View Independent Constraints for Coplanar Motion

In this Section, we derive view independent constraints on a configuration of points having independent uniform linear velocity or acceleration such that all the points lie on the same plane for the entire duration. We parameterize the projection of the point and then compute an invariant for the set of points. The point's projection has time as a parameter and so we have a time varying invariant - a time varying constraint.

5.2.1 Uniform Linear Velocity

Suppose that we have a world point P that moves with uniform linear velocity as per Equation 2.3. Let this motion be viewed by a projective camera, represented by the camera matrix $\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix}$ Therefore (x, y) - the position of the projection of the point at any time t is given by

$$x(t) = \frac{(m_{11}I_x + m_{12}I_y + m_{13}I_z + m_{14}) + t(m_{11}U_x + m_{12}U_y + m_{13}U_z)}{(m_{31}I_x + m_{32}I_y + m_{33}I_z + m_{34}) + t(m_{31}U_x + m_{32}U_y + m_{33}U_z)}$$
$$y(t) = \frac{(m_{21}I_x + m_{22}I_y + m_{23}I_z + m_{24}) + t(m_{21}U_x + m_{22}U_y + m_{23}U_z)}{(m_{31}I_x + m_{32}I_y + m_{33}I_z + m_{34}) + t(m_{31}U_x + m_{32}U_y + m_{33}U_z)}$$

or alternatively,

$$x(t) = \frac{A + tB}{E + Ft}$$
$$y(t) = \frac{C + tD}{E + Ft}$$

where,

$$\begin{aligned} A &= (m_{11}I_x + m_{12}I_y + m_{13}I_z + m_{14}) \,, \\ B &= (m_{11}U_x + m_{12}U_y + m_{13}U_z) \,, \\ C &= (m_{21}I_x + m_{22}I_y + m_{23}I_z + m_{24}) \,, \\ D &= (m_{21}U_x + m_{22}U_y + m_{23}U_z) \,, \\ E &= (m_{31}I_x + m_{32}I_y + m_{33}I_z + m_{34}) \text{ and} \\ F &= (m_{31}U_x + m_{32}U_y + m_{33}U_z) \end{aligned} \tag{5.1}$$

We can set one of the unknowns, say F to 1, to get

$$x(t) = \frac{A + tB}{E + t}$$
$$y(t) = \frac{C + tD}{E + t}$$

Hence, we can parametrize the position of the projection of a point at any time t with *five* unknowns which we can calculate from three time instants since each time instant provides two equations – one in x and one in y.

Can we do better? In the sense that can we parameterize the motion in terms of fewer unknowns? This is what we investigate next.

Parameterization using Lines Perpendicular to the Line of Motion

To achieve parameterization in terms of fewer unknowns than above, we consider the line of motion of the projection, as well as the line perpendicular to the line of motion. Let $(b, -a, d)$ be the line traced out by the projection of a point moving with constant velocity. Therefore, the line perpendicular to this line of motion can be represented by $l(t) = (a, b, c(t))$. (Note that a and b are constant, only the 'constant' term varies with time.)

Let $\mathbf{p}(t)$ be the projection of the world point $(\begin{bmatrix} \mathbf{I} \\ 1 \end{bmatrix} + t \begin{bmatrix} \mathbf{U} \\ 0 \end{bmatrix})$ due to a camera $\mathbf{M}$. Let $l(t)$ be the line perpendicular to the line of motion passing through $p(t)$. Therefore, we have

$$l(t)^T \mathbf{p}(t) = 0$$

Replacing $\mathbf{p}(t)$ with $\mathbf{MP}$

$$\begin{bmatrix} a & b & c(t) \end{bmatrix} \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} I_x + tU_x \\ I_y + tU_y \\ I_z + tU_z \\ 1 \end{bmatrix} = 0$$

On expanding we get,

$$\begin{aligned} &a[m_{11}(I_x + tU_x) + m_{12}(I_y + tU_y) + m_{13}(I_z + tU_z) + m_{14}] + \\ &b[m_{21}(I_x + tU_x) + m_{22}(I_y + tU_y) + m_{23}(I_z + tU_z) + m_{24}] + \\ &c(t)[m_{31}(I_x + tU_x) + m_{32}(I_y + tU_y) + m_{33}(I_z + tU_z) + m_{34}] = 0 \end{aligned}$$

This can be written as

$$a(m_{11}I_x + m_{12}I_y + m_{13}I_z + m_{14}) + at(m_{11}U_x + m_{12}U_y + m_{13}U_z) +$$
$$b(m_{21}I_x + m_{22}I_y + m_{23}I_z + m_{24}) + bt(m_{21}U_x + m_{22}U_y + m_{23}U_z) +$$
$$c(t)(m_{31}I_x + m_{32}I_y + m_{33}I_z + m_{34}) + c(t)t(m_{31}U_x + m_{32}U_y + m_{33}U_z) = 0$$

$\mathbf{I}$ and $\mathbf{U}$ are constants of the world motion, a and b are constants of the motion of the projection, while m_{ij} are camera constants. Therefore,

$$\begin{aligned} L &= a(m_{11}I_x + m_{12}I_y + m_{13}I_z + m_{14}) , \\ M &= a(m_{11}U_x + m_{12}U_y + m_{13}U_z) , \\ N &= b(m_{21}I_x + m_{22}I_y + m_{23}I_z + m_{24}) , \\ O &= b(m_{21}U_x + m_{22}U_y + m_{23}U_z) , \\ P &= (m_{31}I_x + m_{32}I_y + m_{33}I_z + m_{34}) , \text{ and} \\ Q &= (m_{31}U_x + m_{32}U_y + m_{33}U_z) \end{aligned}$$

are constants when analysing the projection of a point in a particular view. Therefore,

$$L + Mt + N + Ot + c(t)P + c(t)tQ = 0$$

On reordering the terms we get

$$(L + N) + (M + O)t + c(t)(P + tQ) = 0$$

Writing $(L + N)$ as R, $(M + O)$ as S, we get

$$R + St + c(t)(P + tQ) = 0$$

Since the above equation is a homogeneous equation, we can fix one of the unknowns, say Q, to be 1. This yields

$$R + St + c(t)(P + t) = 0$$

Solving for $c(t)$ we get

$$c(t) = -\frac{R + St}{P + t} \tag{5.2}$$

Note that in this formulation, we have only 3 unknowns - R, S, and P which we can calculate from three time instants.

Now we can parameterize the projection of a point at any time instant, by taking the cross product of the line of motion $(b, -a, d)$ and the perpendicular line $l(t) = (a, b, c(t))$ to get the $x(t)$ and $y(t)$ coordinates as above. This representation of $x(t)$ and $y(t)$ is similar to the one above, but the number of *effective* unknowns is only 3 instead of 5.

Justification of the Decrease in the Number of Unknowns

Claim: *The number of real unknown parameters of the system remain constant = 7*

In the case, where in we represent the projections of points as

$$x(t) = \frac{A + tB}{E + t}$$
$$y(t) = \frac{C + tD}{E + t}$$

we make 2 measurements - $x(t)$ and $y(t)$ and there are 5 unknowns - $A, B, C, D, andE$ giving us a total of 7 unknown parameters.

When using lines perpendicular to the line of motion, we parameterize the constant term of the lines as

$$c(t) = -\frac{R + St}{P + t} \tag{5.3}$$

We make 4 measurements - $a, b, d, c(t)$ and have 3 unknowns - R, S, and P giving us a total of 7 unknown parameters.

Thus, we are able to parameterize the position of the projection of the point with fewer number of unknowns when using lines perpendicular to the line of motion because we are making more measurements.

Time Varying Invariant

The points of the configuration lie on the same plane during the entire duration, so the various cameras observing the motion, image points on the same plane. Then the various views of the point configuration are related by a projective homography [1]. To express the configuration in a view-independent manner all we need is to use an invariant to projective transformations of 2D like the ones given in [2]. Given a configuration of five points in an image in general position i.e. no three are collinear, we can define an invariant like the cross ratio of areas. Five points parameterized as above can be used to define the invariant which will be a function of time as the points have time as a parameter.

Let the projections of the five points in a view be

$$\begin{aligned} \mathbf{p_i}(t) &= \begin{bmatrix} x_i(t) & y_i(t) & 1 \end{bmatrix} \text{, where} \\ x_i(t) &= \frac{A_i + tB_i}{E_i + t} \\ y_i(t) &= \frac{C_i + tD_i}{E_i + t} \quad 1 \leq i \leq 5 \end{aligned}$$

The number of effective unknowns is only 3 if we make use of the lines perpendicular to the line of motion. We then define the cross ratio of areas of these five points (Equation 1.2) as

$$cr(\mathbf{p_1}, \mathbf{p_2}, \mathbf{p_3}, \mathbf{p_4}, \mathbf{p_5}) = \frac{\triangle_{\mathbf{p_1 p_2 p_5}} \cdot \triangle_{\mathbf{p_3 p_4 p_5}}}{\triangle_{\mathbf{p_1 p_3 p_5}} \cdot \triangle_{\mathbf{p_2 p_4 p_5}}} \ , \tag{5.4}$$

where $\triangle_{\mathbf{p_i p_j p_k}}$ is the area of the triangle formed by points $\mathbf{p_i}, \mathbf{p_j}$, and $\mathbf{p_k}$.

The resultant expression is the ratio of two polynomials of degree 6 in the time parameter t. The number of *effective* unknowns in this expression of the projections of 5 points is only 15 (3 for each of the 5 points). Only three time instants in each view are required to determine this time varying invariant. This is a significant theoretical advancement over the formulation presented in Levin et al [36] that imposes a constraint on the projection of 6 points having 35 unknowns, computing which need 34 time instants.

This result can be easily extended to the case of configurations of points having independent uniform linear accelerations which is done next.

5.2.2 Uniform Linear Acceleration

Let $\mathbf{P}$ be a point moving in the world with uniform linear acceleration as per Equation 2.9, which is imaged by a projective camera. The position of the projection $\mathbf{p} = [x(t), y(t)]$ at any time instant t is given by

$$x(t) = \frac{(m_{11}I_x + m_{12}I_y + m_{13}I_z + m_{14}) + t(m_{11}U_x + m_{12}U_y + m_{13}U_z) + 0.5 * t^2(m_{11}A_x + m_{12}A_y + m_{13}A_z)}{(m_{31}I_x + m_{32}I_y + m_{33}I_z + m_{34}) + t(m_{31}U_x + m_{32}U_y + m_{33}U_z) + 0.5 * t^2(m_{31}A_x + m_{32}A_y + m_{33}A_z)}$$

$$y(t) = \frac{(m_{21}I_x + m_{22}I_y + m_{23}I_z + m_{24}) + t(m_{21}U_x + m_{22}U_y + m_{23}U_z) + 0.5 * t^2(m_{21}A_x + m_{22}A_y + m_{23}A_z)}{(m_{31}I_x + m_{32}I_y + m_{33}I_z + m_{34}) + t(m_{31}U_x + m_{32}U_y + m_{33}U_z) + 0.5 * t^2(m_{31}A_x + m_{32}A_y + m_{33}A_z)}$$

or alternatively,

$$x(t) = \frac{A + Bt + Ct^2}{G + Ht + It^2}$$
$$y(t) = \frac{D + Et + Ft^2}{G + Ht + It^2}$$

where,

$$\begin{aligned}
A &= (m_{11}I_x + m_{12}I_y + m_{13}I_z + m_{14}) , \\
B &= (m_{11}U_x + m_{12}U_y + m_{13}U_z) , \\
C &= 0.5 * (m_{11}A_x + m_{12}A_y + m_{13}A_z) , \\
D &= (m_{21}I_x + m_{22}I_y + m_{23}I_z + m_{24}) , \\
E &= (m_{21}U_x + m_{22}U_y + m_{23}U_z) , \\
F &= 0.5 * (m_{21}A_x + m_{22}A_y + m_{23}A_z) , \\
G &= (m_{31}I_x + m_{32}I_y + m_{33}I_z + m_{34}) , \\
H &= (m_{31}U_x + m_{32}U_y + m_{33}U_z) , \text{ and} \\
I &= 0.5 * (m_{31}A_x + m_{32}A_y + m_{33}A_z)
\end{aligned}$$

We can arbitrarily set one of the unknowns I to 1, to get the parameterized form

$$x(t) = \frac{A + Bt + Ct^2}{G + Ht + t^2}$$
$$y(t) = \frac{D + Et + Ft^2}{G + Ht + t^2} \tag{5.5}$$

Hence, we can parameterize the projections of a point moving with uniform linear acceleration in the world with 8 unknowns. Each time instant provides two equations – one each in $x(t)$ and $y(t)$. Therefore, we need 4 time instants to identify all the unknown parameters. Like in the case of uniform linear velocity, considering lines perpendicular to the line of motion, we can parameterize the motion of the projection with fewer unknowns.

Parameterization using Lines Perpendicular to the Line of Motion

Let $(b, -a, d)$ be the line traced out by the projection of a point moving with constant velocity. Therefore, the line perpendicular to this line of motion can be represented by $l(t) = (a, b, c(t))$. Only the 'constant' term in the line $l(t)$ varies with time. Proceeding as in the case of uniform linear velocity, we get

$$c(t) = -\frac{P + Qt + Rt^2}{S + Tt + t^2} \tag{5.6}$$

In this formulation, we have only 5 unknowns - P, Q, R, S, and T which we can compute from 5 time instants. Note that this expression has fewer unknowns than the parametric representation in Equation 5.5 but the number of time instants needed to compute them is greater.

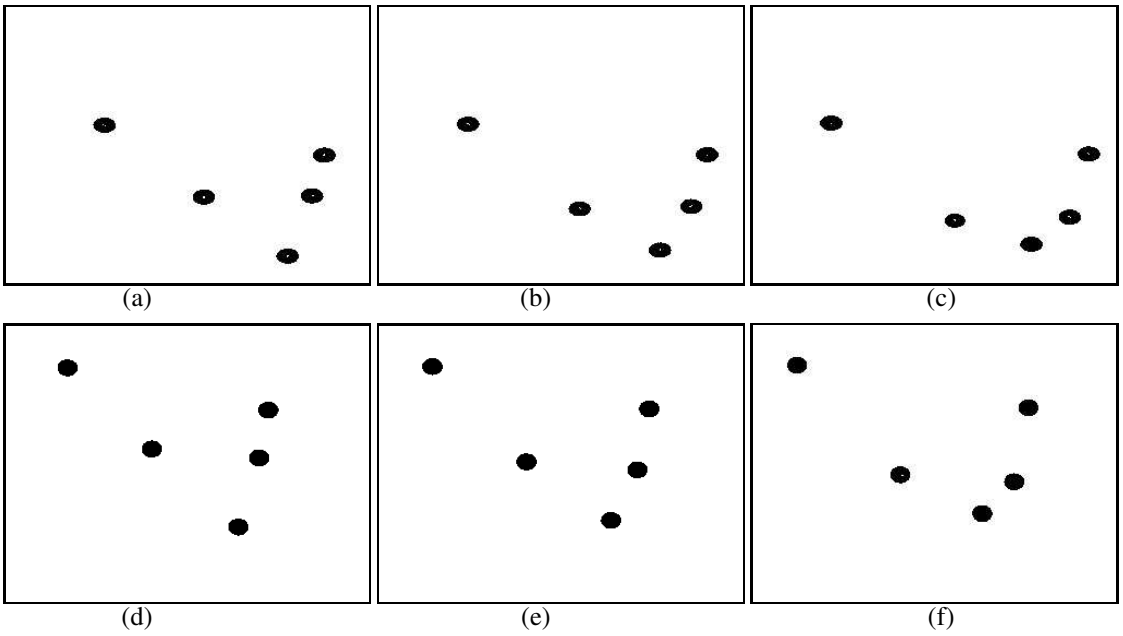

Figure 5.1: Three frames each of two views of a configuration of points moving with independent uniform linear velocity

Now we can parameterize the projection of a point at any time instant, by taking the cross product of the line of motion $(b, -a, d)$ and the perpendicular line $l(t) = (a, b, c(t))$ to get the $x(t)$ and $y(t)$ coordinates as above.

We can now define an invariant like the cross ratio of areas we defined in Section 5.2.1. The invariant will be of order 12 in the time parameter t and have only 25 unknowns (5 for each point). This is the first time that such time varying invariants have been defined for the case of a configuration of points moving with independent uniform linear acceleration.

5.2.3 Experiments

To test the validity of these constraints we conducted a number of experiments. Figure 5.1 shows three frames each of two views of a configuration of 5 points moving with independent uniform linear velocities. The unknown parameters P, R, and S were computed in each view.

Equation 5.2 can be used to determine the time instant given $c(t)$

$$t = -\frac{R + Pc(t)}{S + c(t)} \tag{5.7}$$

The time instants computed using Equation 5.7 were found to be correct in all cases. The time varying invariant was also computed, which was found to be valid in both views.

Similar tests were conducted on configurations having points moving with independent uniform linear accelerations. To determine the time instants from the $c(t)$ component of the line perpendicular to the line of motion, we would have to solve a quadratic equation in time t obtained from Equation 5.6. The time instants computed in this manner were

found to be correct in all cases. Successful tests were also conducted to compute the corresponding time varying invariant.

Closure

In Chapter 2, we had derived view-independent constraints on configurations of points moving with independent uniform linear velocities or accelerations when the projection model is affine. In this chapter we have first parameterized the projection of a point when the camera model is projective. Then for the case when the motion of the configuration is restricted to a plane, we used the notion of the existence of a projective homography between the various views of the configuration, to define a time varying invariant. Our novel parameterizations yield invariants whose computation require fewer number of points and time instants than past attempts reported in literature. Experimental validation of these constraints has also been presented.

In these chapters, we have presented a number of constraints on moving point configurations. These constraints can contribute positively to a number of applications like recognition, determining view consistency, etc. In the next chapter, we analyse the application of these constraints to solving the problem of aligning frames of synchronized videos.

Chapter 6

Alignment of Frames of Synchronized Videos

Multiple independent views of a dynamic event can be obtained using multiple video cameras. The multiview algebraic relations are then satisfied between the corresponding points of the views of the *same* time instant, provided the videos are synchronized to a common video signal. Using a still-camera analogy, synchronization ensures that the "shutters" of all cameras are opened at the same time instant. Thus, the visual world is sampled at the same time instants by all views. However, aligning the discretized time axes of each video to a common sequence so that the specific time instants in different views can be identified is a non-trivial task even for synchronized videos. This is the *frame-alignment problem* for multiple views. Aligning the frames in this manner is the first step of all Computer Vision algorithms using multiple views. Figures 6.1,6.2, and 6.3 show three situations of watching an event from different and wide-apart viewpoints. A few solutions to this problem have appeared in the literature [19, 22].

In this report, we have presented a number of constraints on the projections of a configuration of points. Many of the constraints have time as a parameter. Thus, given a configuration and the constraint parameters, the time instant can be identified, enabling us to achieve alignment of frames of synchronized videos.

Organisation of the chapter In Section 1, we analyse frame alignment using constraints on affine projections of configurations, while in Section 2, frame alignment in a projective framework is presented.

6.1 Affine Cameras

In this Section, we show how the constraints on the affine projections of points that we have derived in previous chapters can be used to achieve frame alignment. In this discussion, we will assume that we have two synchronized video sequences $\mathbf{A}$, $\mathbf{B}$ and we are interested in aligning them.

6.1.1 Using Configurations having Uniform Linear Motion

In Chapter 2, we had studied view-independent constraints on the projections of points moving with independent uniform linear velocities or accelerations. The time-independent constraints do not have a time parameter, but the time-dependent ones do and can be used to determine the time instant given the configuration and the view-independent

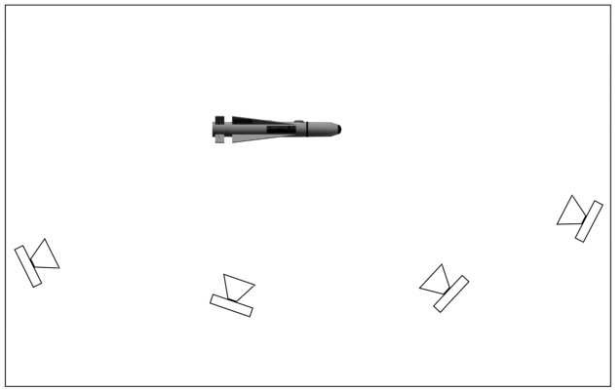

Figure 6.1: A set of ground stations observing a ballistic motion

constraint parameters. Time-dependent constraints on the velocities of the projections of points moving with independent uniform linear accelerations in the world have been derived (Equations 2.10, 2.11).

If we can identify points moving with constant acceleration, Equations 2.10 and 2.11 would hold for all views for the same β's. The time t can be replaced with the frame number. From the image velocities of the projections of 4 points in 8 frames in view $\mathbf{A}$, β's that characterize the point configuration can be computed. To achieve alignment, we need to identify the corresponding frame k in view $\mathbf{A}$ for the frame j in view $\mathbf{B}$ which would give the shift required for alignment as $(k - j)$. Let the image velocities of the projections of the four points in view $\mathbf{B}$ at time instant j be (v_{ixj},v_{iyj}), $1 \leq i \leq 4$. We have

$$\begin{aligned} (\ \beta_0 + \beta_1 k + \beta_2 k^2 + \beta_3 k^3\ ) v_{1xj} + \\ (\ \beta_4 + \beta_5 k + \beta_6 k^2 + \beta_7 k^3\ \ ) v_{2xj} + \\ (\beta_8 + \beta_9 k + \beta_{10} k^2 + \beta_{11} k^3) v_{3xj} + \\ (\ \beta_{12} + \beta_{13} k + \beta_{14} k^2 + \beta_{15} k^3) v_{4xj} \quad = \quad 0 \end{aligned} \tag{6.1}$$

To determine the frame number k, we can solve for the roots of a cubic polynomial of the form

$$\chi_0 k^3 + \chi_1 k^2 + \chi_2 k + \chi_3 = 0 \tag{6.2}$$

where $\chi_0 = (\beta_3 v_{1xj} + \beta_7 v_{2xj} + \beta_{11} v_{3xj} + \beta_{15} v_{4xj})$, $\chi_1 = (\beta_2 v_{1xj} + \beta_6 v_{2xj} + \beta_{10} v_{3xj} + \beta_{14} v_{4xj})$, $\chi_2 = (\beta_1 v_{1xj} + \beta_5 v_{2xj} + \beta_9 v_{3xj} + \beta_{13} v_{4xj})$, and $\chi_3 = (\beta_0 v_{1xj} + \beta_4 v_{2xj} + \beta_8 v_{3xj} + \beta_{12} v_{4xj})$. (Note that the same exercise can be done using v_y values.)

To determine the validity of this technique for frame alignment, tests were conducted on a number of scenes. A simulated explosion of the teapot, with the various fragments flying off in different directions with different but constant acceleration, was used for experimentation. Figure 2.2 shows a few frames from the image sequences of the exploding tea pot from two different view points. Tests were carried out to evaluate the applicability of this technique for frame alignment, by varying the starting point of the second video sequence. In all the cases the proper shift value was recovered. In one experiment, the two sequences were misaligned by seven frames. The fit-error graph obtained on searching over the possible range of shifts, for aligning the sequences is shown in Figure 6.5. We can see that the

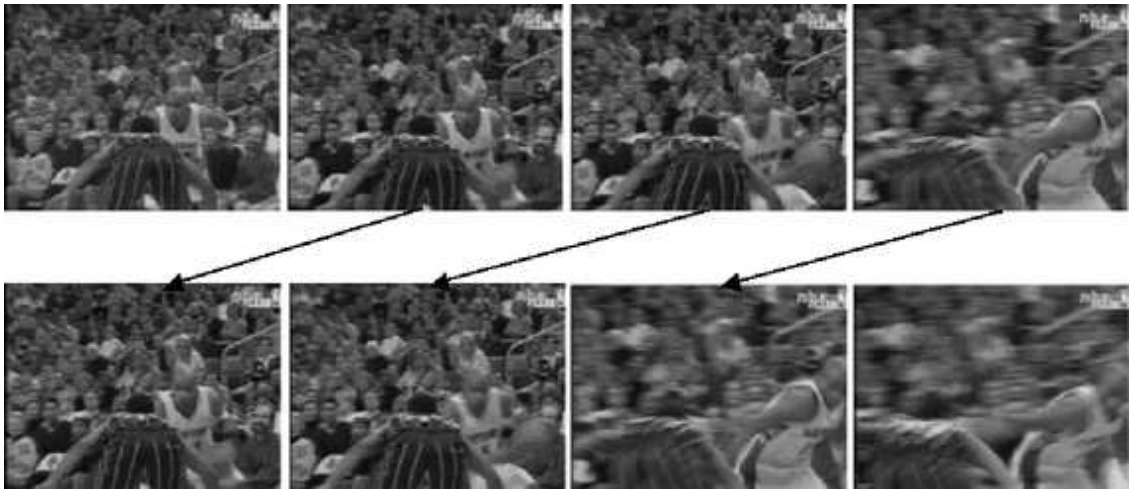

Figure 6.2: Two videos of a sports event and their alignment

fit-error falls as we approach the correct shift value, attains a minimum at the correct shift value and then increases again as we move away from the correct shift value.

We next analyse the case when a point moves on an arbitrary trajectory confined to a plane.

6.1.2 Using a Point having Arbitrary Planar Motion

It has been shown in Section 3.2 that the phases of $\kappa'_p(B)$ and $\kappa'_p(A)$ differ by an amount proportional to the required shift for alignment λ_B and the differential frequency $(k-p)$ (Equation 3.9). Therefore, the ratio $\frac{\kappa'_p(B)}{\kappa'_p(A)}$ will be a complex sinusoid $ce^{-j2\pi\lambda_B(k-p)/N}$. The value of λ_B can be computed from the inverse Fourier transform of the quotient series. Figure 6.6 shows the inverse Fourier transform of the quotient series $\frac{\kappa'_p(B)}{\kappa'_p(A)}$ which has a peak at the proper shift value of 7.

6.1.3 Using Deforming Contours

We will next see how we can use time-dependent constraints on deforming contours to achieve frame alignment. The problem that we would like to address is that if we are given two views of a deforming contour, can we determine the time instant in one view that corresponds to the given view of the deforming contour.

When we can perform tracking only in the reference view, the value of κ is given by Equation 4.7

$$\kappa_t^B[k] = det(\mathbf{A}^l)(\alpha_1[k] + \alpha_2[k]t + \alpha_3[k]t^2)$$

Normalizing $\kappa_t^B[k]$ with respect to a fixed frequency (say p) gives

$$\frac{\kappa_t^B[k]}{\kappa_t^B[p]} = \frac{\alpha_1[k] + \alpha_2[k]t + \alpha_3[k]t^2}{\alpha_1[p] + \alpha_2[p]t + \alpha_3[p]t^2}$$

The above is a quadratic in time t, solving for which, we can find the time instant (frame number) in the reference frame $\mathbf{A}$ corresponding to the frame in view $\mathbf{B}$. This constraint was also evaluated using simulated data.

Figure 6.3: Observing an event using multiple cameras (Courtesy Keck Laboratory, University of Maryland)

Till now we have designed frame alignment techniques based on the constraints on the affine projections of a configuration of points. We now look at using the constraints developed for projective cameras in Chapter 5 to achieve frame alignment.

6.2 Projective Cameras

We consider the scenario when the points in the scene move with independent uniform linear velocities. We assume that we have three synchronized video sequences $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ of which $\mathbf{A}$ and $\mathbf{B}$ are time aligned as well and we are interested in aligning them with $\mathbf{C}$.

The constant term of the line perpendicular to the line of motion is then parameterized as (Equation 5.2)

$$c(t) = -\frac{R + St}{P + t}$$

Given the constant term $c(t)$, we can find the time instant as (Equation 5.7)

$$t = -\frac{R + Pc(t)}{S + c(t)}$$

Now if we want to find the time instant (frame) in view $\mathbf{C}$ that corresponds to the frame k in views $\mathbf{A}$ and $\mathbf{B}$, we first determine the lines perpendicular to the line of motion at time k in the views $\mathbf{A}$ and $\mathbf{B}$. Then using the Tri-linear Tensor we transfer the perpendicular lines in $\mathbf{A}$ and $\mathbf{B}$ to view $\mathbf{C}$ using Equation 1.12. Now we have the corresponding line perpendicular to the line of motion in view $\mathbf{C}$. Using Equation 5.7 we can then find the corresponding time instant in view $\mathbf{C}$. This strategy was evaluated on simulated scenes in which the proper time instants in view $\mathbf{C}$ were obtained.

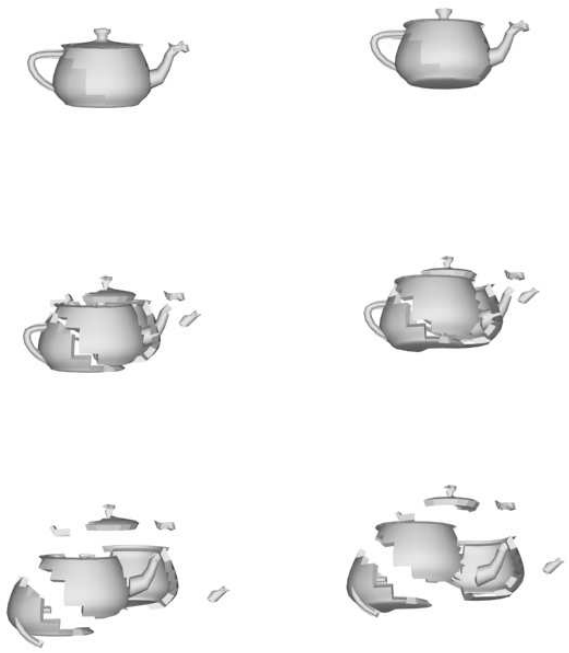

Figure 6.4: Two image sequences of an exploding pot

Closure

In this chapter, we have applied the time dependent constraints derived in previous chapters to designing techniques for alignment of frames of synchronized videos. Both affine and projective camera models were considered. The formulations were validated through a experiments some of which have been presented here.

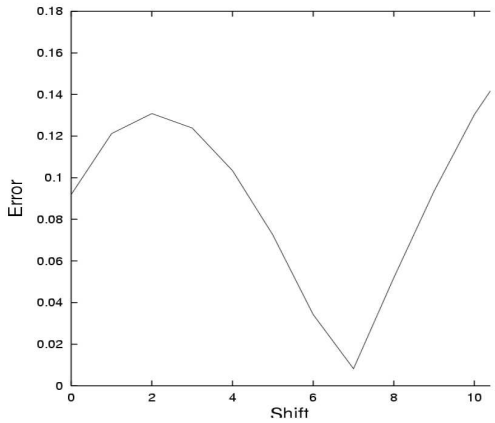

Figure 6.5: Alignment of image sequences by searching over the range of possible shifts (See text for more details)

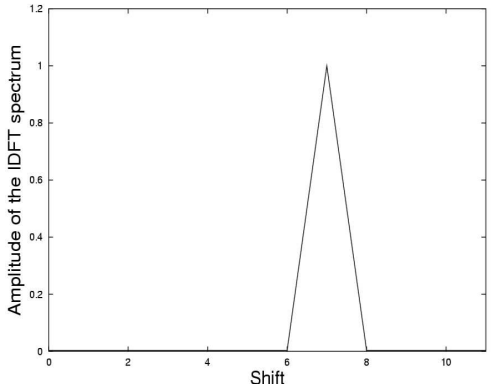

Figure 6.6: Alignment determination in the Fourier Domain. (See text for more details)

Chapter 7

Conclusions

In course of this work, we have studied constraints on the projections of configurations of points in motion. We first outlined the philosophy of eliminating camera parameters to arrive at view independent constraints for static configurations of points. This was then extended to derive time-dependent and time-independent view-independent constraints on the projections of configurations of points moving with uniform linear velocities or accelerations. Such modeling of a configuration can be used to model non-rigid motion in some cases. We then made the observation that a point in motion traces out a contour in the world. The contour traced out in the image by the projection of the point over time is a projection of the contour in the world. The problem of analysing the motion of the point can then be posed as a problem in contour analysis which is fairly well researched. This enables us to accommodate for truly non-rigid motion. Shape constraints were then combined with motion constraints to derive novel recognition strategies for deforming contours. In addition to recognition of configurations, we have outlined the problem of alignment of frames of synchronized videos and showed how the constraints that we have derived can contribute positively to such an application.

We have improved upon a number of theoretical results and explored new directions in the study of motion in multiple views, in addition to designing new techniques for recognition and video frame alignment.

Bibliography

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

[2] J. Mundy and A. Zisserman, *Geometric Invariances in Computer Vision*. MIT Press, 1992.

[3] R. Gonzalez and R. Woods, *Digital Image Processing*. Addison Wesley Longman, 1992.

[4] A. K. Jain, *Fundamentals of Digital Image Processing*. Prentice-Hall, 1989.

[5] E. Kruppa, "Zur ermittlung eines objecktes aus zwei perspektiven mit innerer orientierung," *Sitz.-Ber.Akad.Wiss., Math.Naturw.*, pp. 1939–1948, 1913.

[6] H. C. Longuet-Higgins, "A Computer Algorithm for Reconstructing a Scene from two Projections," *Nature*, vol. 293, pp. 133–135, 1981.

[7] O. Faugeras, *Three Dimensional Computer Vision*. MIT Press, 1992.

[8] J. Weng, T. S. Huang, and N. Ahuja, "Motion and Structure from Line Correspondences: Closed-form Solution, Uniqueness and Optimization," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, pp. 318–336, March 1992.

[9] M. Spetsakis and J. Aloimonos, "Structure from Motion using Line Correspondences," *International Journal of Computer Vision*, vol. 4(3), pp. 171–183, 1990.

[10] A. Shashua, "Trilinear tensor: The Fundamental Construct of Multiple-view Geometry and its applications," *Int. Worskshop on AFPAC*, 1997.

[11] R. Hartley, "Lines and points in three views: An integrated approach," *Proc. ARPA Image Understanding Workshop*, 1994.

[12] O. Faugeras and B. Mourrain, "On the geometry and algebra of the point and line corrrespondences between n images," *Proc. International Conference on Computer Vision*, 1995.

[13] B. Triggs, "Matching Constraints and the Joint Image," *International Conference on Computer Vision*, pp. 338–343, 1995.

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

[15] A. Shashua, "Algebraic Functions for Recognition," *IEEE Tran. Pattern Anal. Machine Intelligence*, vol. 16, pp. 778–790, 1995.

[16] D. P. Huttenlocher and S. Ullman, "Object Recognition using Alignment," *Proc. International Conference on Computer Vision*, pp. 102–111, 1987.

[17] S. Ullman, "Aligning pictorial descriptions: An approach to object recognition," *Cognition*, vol. 32, pp. 193–254, 1989.

[18] S. Ullman and R. Basri, "Recognition by Linear Combination of Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 992–1006, 1991.

[19] Y. Caspi and M. Irani, "Alignment of non-overlapping sequences," *ICCV*, vol. 2, pp. 76–83, 2001.

[20] Y. Caspi and M. Irani, "A step towards sequence-to-sequence alignment," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 682–689, 2000.

[21] G. Stein, "Tracking from multiple view points: Self-calibration of space and time," *DARPA IU Workshop*, pp. 1037–1042, 1998.

[22] Kuthirumal Sujit, C. V. Jawahar, and P. J. Narayanan, "Video frame alignment in multiple views," *International Conference on Image Processing*, 2002.

[23] S. Avidan and A. Shashua, "Novel view synthesis by cascading trilinear tensors," *IEEE Transactions on Visualization and Computer Graphics*, vol. 4, pp. 293–306, 1998.

[24] S. Chen, "Quicktimevr–an image-based approach to virtual environment navigation," *SIGGRAPH*, 1995.

[25] T. Werner, R. Hersch, and V. Hlavac, "Rendering real-world objects using view interpolation," *Proc. International Conference on Computer Vision*, 1995.

[26] S. Laveau and O. Faugeras, "3-d scene representation as a collection of images," *Proc. International Conference on Pattern Recognition*, 1994.

[27] L. McMillan and G. Bishop, "Plenoptic modeling: An image-based rendering system," *SIGGRAPH*, 1995.

[28] A. W. Fitzgibbon and A. Zisserman, "Automatic camera recovery for closed or open image sequences," *Proc. European Conference on Computer Vision*, pp. 311–326, 1998.

[29] D. Nister, "Automatic dense reconstruction from uncalibrated video sequences," *KTH*, 2001.

[30] R. Hartley, N. Dano, and R. Kaucic, "Plane-based projective reconstruction," *International Conference on Computer Visualization*, pp. 420–427, 2001.

[31] A. W. Fitzgibbon, G. Cross, and A. Zisserman, *Automatic 3D Model Construction for Turn-Table Sequences*. Springer, 1998.

[32] R. Vidal, Y. Ma, S. Hsu, and S. Sastry, "Optimal motion estimation from multiview normalized epipolar constraint," *Proc. International Conference on Computer Vision*, 2001.

[33] C. Tomasi and T. Kanade, "Shape and motion from image streams: A factorization method," *Technical report, Carnegie Mellon University CMU-CS-92-104*, 1992.

[34] B. Triggs, "Factorization methods for projective structure and motion," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 845–851, 1996.

[35] S. Carlsson, "Recognizing walking people," *European Conference on Computer Vision*, June 2000.

[36] A. Levin, L. Wolf, and A. Shashua, "Time-varying Shape Tensors for Scenes with Multiple Moving Points," *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[37] M. Isard and A. Blake, "Condensation–conditional density propagation for visual tracking," *International Journal on Computer Vision*, vol. 28, pp. 5–28, 1998.

[38] L. Torresani, D. B. Yang, E. J. Alexander, and C. Bregler, "Tracking and modeling non-rigid objects with rank constraints," *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[39] D. Reynard, A. Wildenberg, A. Blake, and J. Marchant, "Learning dynamics of complex motions from image sequences," *Proc. European Conference on Computer Vision*, pp. 357–368, 1996.

[40] T. Pavlidis, *Structural Pattern Recognition*. Springer-verlag, 1977.

[41] C. Zahn and R. Roskies, "Fourier Descriptors for Planar curves," *IEEE Trans. Comput.*, vol. C-21, 1972.

[42] K. Arbter, W. Snyder, H. Burkhardt, and G. Hirzinger, "Application of Affine-Invariant Fourier Descriptors to Recognition of 3d Objects," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, 1990.

[43] S. Kuthirummal, "Multiview geometry in transform domain and its applications," *Undergraduate Thesis*, 2002.

[44] Sujit Kuthirummal, C. V. Jawahar, and P. J. Narayanan, "Planar Shape Recognition across Multiple Views," *International Conference on Pattern Recognition*, 2002.

[45] Sujit Kuthirummal, C. V. Jawahar, and P. J. Narayanan, "Multiview constraints for recognition of planar curves in fourier domain," *Indian Conference on Computer Vision, Graphics and Image Processing*, 2002.

[46] S. Kuthirummal, C. Jawahar, and P. Narayanan, "Fourier domain representation of planar curves for recognition in multiple views," *To appear in Pattern Recognition*, 2003.

[47] M. P. Kumar, S. Goyal, S. Kuthirummal, C. Jawahar, and P. Narayanan, "Discrete contours in multiple views: Approximation and recognition," *To appear in Image and Vision Computing*, 2003.

[48] K. J. Dana and S. Nayar, "Correlation model for 3d texture," *International Conference on Computer Vision*, 1999.

[49] B. V. Ginneken, J. J. Koenderink, and K. J. Dana, "Texture histograms as a function of irradiation and viewing direction," *International Journal of Computer Vision*, vol. 31(2/3), pp. 169–184, 1999.

[50] F. Schaffalitzky and A. Zisserman, "Viewpoint invariant texture matching and wide baseline stereo," 2001.

[51] L. Shapiro, A. Zisserman, and M. Brady, "3D Motion Recovery via Affine Epipolar Geometry," *IJCV*, vol. 16(2), pp. 147–182, 1995.

[52] B. M. Bennet, D. D. Hoffman, and C. Prakash, "Recognition Polynomials," *Journal of the Optical Society of America*, vol. 10, pp. 759–764, 1993.

[53] Sujit Kuthirummal, C. V. Jawahar, and P. J. Narayanan, "Algebraic constraints on moving points in multiple views," *Indian Conference on Computer Vision, Graphics and Image Processing*, 2002.

[54] T. Crimmins, "A complete set of Fourier descriptors for two dimensional shapes," *IEEE Trans. Syst. Man. Cybern.*, vol. SMC-12, pp. 195–201, 1982.

[55] G. Granlund, "Fourier preprocessing for hand printed character recognition," *IEEE Trans. Comput.*, vol. C-21, 1972.

[56] R. Hartley, "Lines and points in three views and the trifocal tensor," *IJCV*, vol. 22, pp. 125–140, 1997.

[57] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Handbook of Automated Computation*, vol. 2, pp. 134–151, 1971.

[58] I. Cohen, N. Ayache, and P. Sulger, "Tracking points on deformable objects using curvature information," *European Conference on Computer Vision*, 1992.

[59] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes:active contour models," *International Journal on Computer Vision*, vol. 1(4), pp. 321–331, 1988.

[60] J. Tsukumo, "Handprinted kanji character recognition based on flexible template matching," *International Conference on Pattern Recognition*, pp. 483–486, 1992.

[61] R. Basri, L. Costa, D. Geiger, and D. Jacobs, "Determining the similarity of deformable shapes," *IEEE Workshop on Physics-Based Modelling in Computer Vision*, pp. 135–143, 1995.

[62] K. Lai and C. R, "Deformable contours:modeling and extraction," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 601–608, 1994.

[63] D. Terzopoulos and D. Metaxus, "Dynamic 3d models with local and global deformations:deformable superquadrics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13(7), pp. 703–714, 1991.

Acknowledgements

I would like to thank my supervisors Dr. C.V. Jawahar and Dr. P.J. Narayanan for their guidance and support. I am also grateful to the members of the Centre for Visual Information Technology (CVIT) for their help and stimulating company.