MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Statistical Methods and Models for Video-Based Tracking, Modeling, and Recognition

Rama Chellappa, Aswin Sankaranarayanan, Ashok Veeraraghavan, Pavan Turaga

TR2010-009 February 2010

Abstract

Computer vision systems attempt to understand a scene and its components from mostly visual information. The geometry exhibited by the real world, the influence of material properties on scattering of incident light, and the process of imaging introduce constraints and properties that are key to solving some of these tasks. In the presence of noisy observations and other uncertainties, the algorithms make use of statistical methods for robust inference. In this monograph, we highlight the role of geometric constraints in statistical estimation methods, and how the interplay of geometry and statistics leads to the choice and design of algorithms. In particular, we illustrate the role of imaging, illumination, and motion constraints in classical vision problems such as tracking, structure from motion, metrology, activity analysis and recognition, and appropriate statistical methods used in each of these problems.

Foundations of Trends in Signal Processing

Foundations and Trends® in
Signal Processing
Vol. 3, No. 1 (2009) 1–150

DOI: 10.1561/2000000007

now
the essence of knowledge

Statistical Methods and Models for Video-Based Tracking, Modeling, and Recognition

By Rama Chellappa, Aswin C. Sankaranarayanan, Ashok Veeraraghavan and Pavan Turaga

Contents

Foundations and Trends® in
Signal Processing
Vol. 3, No. 1 (2009) 1–150

DOI: 10.1561/2000000007

now
the essence of knowledge

Statistical Methods and Models for Video-Based Tracking, Modeling, and Recognition

Rama Chellappa[1], Aswin C. Sankaranarayanan[2], Ashok Veeraraghavan[3] and Pavan Turaga[4]

[1] *Department of Electrical and Computer Engineering, UMIACS, at University of Maryland, College Park, MD, rama@cfar.umd.edu*
[2] *Department of Electrical and Computer Engineering, Rice University, Houston, TX, saswin@rice.edu*
[3] *Mistubishi Electric Research Laboratory, Cambridge, MA, veerarag@merl.com*
[4] *Department of Electrical and Computer Engineering, UMIACS, at University of Maryland, College Park, MD, pturaga@cfar.umd.edu*

Abstract

Computer vision systems attempt to understand a scene and its components from mostly visual information. The geometry exhibited by the real world, the influence of material properties on scattering of incident light, and the process of imaging introduce constraints and properties that are key to solving some of these tasks. In the presence of noisy observations and other uncertainties, the algorithms make use of statistical methods for robust inference. In this monograph, we highlight the role of geometric constraints in statistical estimation methods, and how the interplay of geometry and statistics leads to the choice and

design of algorithms. In particular, we illustrate the role of imaging, illumination, and motion constraints in classical vision problems such as tracking, structure from motion, metrology, activity analysis and recognition, and appropriate statistical methods used in each of these problems.

1

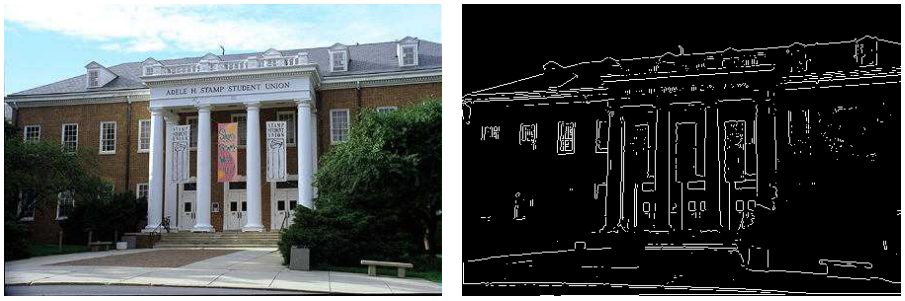
Introduction

The goal of computer vision is to enable machines to see and interpret the world. Computer vision algorithms use input from one or more still images or video sequences that are related in a specific manner. The distribution of intensities and their spatial and temporal arrangements in an image or a video sequence contains information about the identity of objects, their reflectance properties, scene structure, and objects in the scene. However, this information is buried in images and video sequences that make it challenging to infer. One of the fundamental reasons for this difficulty occurs because mapping from the 3D scene to 2D images is not generally invertible. Most traditional computer vision algorithms make appropriate assumptions about the nature of the 3D world and acquisition of images and videos, so the problem of inferring scene properties of interest from sensed data becomes recoverable and analytically tractable.

Within this context, reasonably accurate yet simple geometric models of scene structure (planar scene, etc.), scene illumination (point source), surface properties (Lambertian, Phong, etc.), imaging structure (camera models) serve critical roles in the design of inference algorithms. Moreover, images and video sequences obtained using imaging devices are invariably corrupted by noise. Common noise

sources in the imaging system are due to shot noise, thermal noise, etc. Inference in this noisy environment is further complicated by the inherent errors in physical modeling. Real surfaces are never truly Lambertian, real cameras are never truly perspective, illumination in a scene is never a point light source, nevertheless inference algorithms make these assumptions in order to make the problem tractable. In addition, motion of objects in a scene could complicate the recovery of scene and object properties due to blur, occlusion, etc. Therefore, it becomes important that the developed inference algorithms can cope with varying sources of error.

To illustrate these sources of error, let us consider the following simple application. Suppose we are interested in designing a robot that can localize and identify the entrances to buildings (see Figure 1.1(a)). To begin, we first define a 'model' of an entrance. For computational tractability, we assume the edges of the entrance form a rectangle. Now, given an image containing the entrance, we might choose to use an edge detector or a corner detector to extract features. Due to image-noise, occlusions, and shadows, the features may not exactly correspond to edge locations. With these noisy feature locations, we proceed to fit two sets of parallel lines, where the lines from different sets are perpendicular to each other. Consider the edge figure in Figure 1.1(b). Finding the set of points corresponding to the entrance and grouping them into a rectangle comprises a combinatorial optimization problem. Suppose



(a) (b)

Fig. 1.1 Fitting a rectangle to an entrance. Various sources of error arise here – feature points are noisy, grouping of the points into a rectangle is a challenge, and a rectangle is not an accurate model for the entrance.

we obtain a solution to this optimization problem, perhaps by using the Hough transform. The final error in fit would have occurred due to noisy measurements, the difficulty in solving the constrained optimization problem, and the error in modeling itself, since the entrance does not appear as a rectangle due to perspective effects. The error would become even worse when the viewing angle moves further from frontal, or if shadows are present, etc.

As this example illustrates, computer vision algorithms involve the interplay between geometric constraints that arise from models of the scene. Inference makes assumptions about the imaging devices and about appropriate statistical estimation techniques that can contend with varying sources of error. This tutorial attempts to re-examine and present several computer vision techniques accordingly.

The acceptance of statistical methods in computer vision has been slow and steady. In the early days of the field, the understanding of the geometrical aspects of the problem was given much attention. When uncertainties due to noise and other errors had to be taken into account, and when massive sensor data became available, the infusion of statistical methods was inevitable. Statistical models and methods entered into computer vision through image models. Non-causal models were first introduced in the analysis of spatial data by Whittle [222]. Subsequently, in the 1960s and 1970s, Markov random fields (MRFs) were discussed in statistical [16, 169] and signal processing literature [223]. In the 1980s, statistical methods were introduced primarily for image representation; thus MRFs [37, 48, 111] and other non-causal representations [38, 111, 112] were suggested for images. This enabled the formulation of problems such as image estimation and restoration [76], and texture analysis (synthesis, classification, and recognition) [44, 51] as maximum a posteriori estimation problems. Appropriate likelihood expressions and prior probability density functions were used to derive the required posterior probability density. Nevertheless, the maximum of the posterior probability density functions did not always yield a closed form expression, requiring techniques such as simulated annealing [76, 119].

The introduction of simulated annealing techniques could be considered a seminal moment as it opened a whole new class of sampling

approaches for synthesis and segmentation of textured images [138] and other early vision problems. Simulated annealing techniques were followed by techniques such as mean field annealing [20], iterated conditional mode [17], and maximum posterior marginal [139]. These techniques are now part and parcel of computer vision algorithms. It is worth noting that MRFs and conditional random fields are making a strong resurgence in graphics and machine learning literature.

Applications of Monte Carlo Markov chain techniques for non-linear tracking problems have also been studied [80]. Since the introduction of the CONDENSATION algorithm in 1996 [97], numerous papers have discussed appearance, shape, and behavior-encoded particle filter trackers. Robust estimation methods offer another statistical area that has received attention in the computer vision literature. Many problems such as fitting lines, curves, and motion models relied on least square fitting techniques which are quite sensitive to the presence of outliers. Since the early 1980s, the use of RANSAC [68, 140], M-estimators [95], and least median square estimators [170] has become valuable in all model fitting problems, including fitting moving surfaces and objects to the optical flow generated by them. Discussions of robust estimation with applications in computer vision can be found in Meer et al. [140].

One of the recurring issues in the development of computer vision algorithms is the need to quantify the quality of the estimates. Haralick [87] pioneered this area. In the classical problem of estimating the 3D structure of a scene from motion cues, which is termed as the 'structure from motion' (SfM) problem, one would like to compute the lower bounds on the variances of the motion and structure estimates. Similar needs arise in camera calibration, pose estimation, image alignment, tracking, and recognition problems. A time-tested approach in statistics — the computation of Cramer–Rao bounds [166] and their generalizations — has been adopted for some computer vision problems.

The exploitation of statistical shape theory for object recognition in still images and video sequences has also been studied intensively since the 1980s. In particular, Cooper and collaborators [28, 27] have developed several algorithms based on Bayesian inference techniques for the object recognition problem. Introduction of statistical inference techniques on manifolds which host various representations used

in shape, identity, and activity recognition problems is garnering a lot of interest [195].

Kanatani pioneered statistical optimization under the constraints unique to vision problems. His books explore the use of group theoretical methods [108] and statistical optimization [109] in image understanding and computer vision. In particular, Kanatani [109] explores parametric fitting under relationships such as coplanarity, collinearity, and epipolar geometry, with focus on the bounds on the estimate's accuracy. Kanatani also explored the idea of *geometric correction* of data to make them satisfy geometric constraints.

Finally, we will be grossly remiss, if we do not acknowledge Prof. Ulf Grenander, who created the area of probabilistic and statistical approaches to pattern analysis and computer vision problems. His series of books [82, 83, 85], and the recent book with Miller [84], have laid the foundations for much of what has been accomplished in statistical inference approaches to computer vision. Prof. Julian Besag's contributions to the development of spatial interaction models [16, 18] and Monte Carlo Markov chain techniques [19] are seminal. Many researchers have developed statistical approaches to object detection and recognition in still images. In particular, Professors David Mumford, Don Geman, Stu Geman, Yali Amit, Alan Yuille, Mike Miller, Anuj Srivastava, Song-Chun Zhu and many others have made significant contributions to statistical approaches to still image-based vision problems. As we are focusing on video-based methods and have page constraints, we are unable to provide detailed summaries of outstanding efforts by the distinguished researchers mentioned above.

1.1 Goals

In this monograph, we will examine several interesting video-based detection, modeling, and recognition problems such as object detection and tracking, structure from motion, shape recovery, face recognition, gait-based person identification, and video-based activity recognition. We will explore the fundamental connections between these different problems in terms of the necessary geometric modeling assumptions used to solve them, and we will study statistical techniques that will

enable robust solutions to these problems. Of course, a host of other image processing applications exist where statistical estimation techniques have found great use. The goal of some of these applications, such as image denoising, image deblurring, and super-resolution, is to recover an image, not 'understand' the scene captured in the image. We therefore will not delve in detail about these applications in this tutorial. An in-depth discussion of some statistical techniques applied to image processing may be found in [78, 99].

Writing this tutorial presented a great challenge. Due to page limitations, we could not include all that we wished. We simply must beg the forgiveness of many of our fellow researchers who have made significant contributions to the problems covered here and whose works could not be discussed.

1.2 Outline

We begin the paper with an in-depth coverage of the various geometric models that are used in imaging in Section 2. Light from illumination sources interacts with materials, reflects off them, and reaches the imaging system. Therefore, it is important to study the reflectance properties of materials. We describe popular models of reflectance, such as the Lambertian and Phong models, and indicate vision applications where such reflectance models find use. Next, we describe popular models for the imaging sensor — the camera. In particular, we provide an in-depth description of the perspective projection model and some of its variants. Image sequences obtained from video cameras are related through scene structure, camera motion, and object motion. We also present models for both image motion (optical flow) and object/camera motion and describe how scene structure, motion, and illumination are coupled in a video.

In Section 3, we describe commonly used statistical estimation techniques such as maximum likelihood and maximum a posteriori, estimators. We also describe robust estimators such as M-estimators. We state the problem of Bayesian inference in dynamical systems and describe two algorithms — the Kalman filter and particle filters — that can

perform Bayesian inference with applications to object tracking and recognition in video sequences.

In Section 4, we develop models for detection, tracking, and recognition in surveillance applications, highlighting the use of appearance and behavioral models for tracking. Section 5 describes an important fundamental problem in computer vision — structure from motion (SfM). SfM techniques study the relationship between the structure of a scene and its observability given motion. In the section, we highlight various approaches to explore this relationship, then use them to estimate both the structure of the scene and the motion. We also discuss Cramer–Rao bounds for SfM methods based on discrete features and optical flow fields.

Section 6 discusses some applications in vision where the parameters of interest lie on a manifold. In particular, we study three manifolds, the Grassmann manifold, Stiefel manifold, and the shape manifold, and show how several vision applications involve estimating parameters that live on these manifolds. We also describe algorithms to perform statistical inference on these manifolds with applications in shape, identity, and activity recognition. Finally, in Section 7, we conclude the monograph with a discussion on future trends.

2

Geometric Models for Imaging

Computer vision, at its core, relies on the interplay of light with environment surfaces that then enters the camera aperture. This is shown graphically in Figure 2.1. Further, dynamics in the scene may be introduced either through rigid or non-rigid motion of scene structures, sensor motion, or changes in illumination. To study and understand the structure of the environment from the captured images, we need to maintain accurate and realistic models of lighting, surface properties, camera geometry, and camera and object motion. In this chapter, we discuss appropriate models for each of these and show when each of these models is appropriate. We will review the basics of imaging, starting from models of surfaces, models for cameras, and models for motion. These models form the basis of much of computer vision, and will appear in the various applications to be considered in the rest of the manuscript.

2.1 Models of Surface Reflectance

The interaction of light with materials is fundamental to imaging, and therefore is important to develop and study models for surfaces. In general, light incident on a surface is absorbed, reflected, scattered, and/or

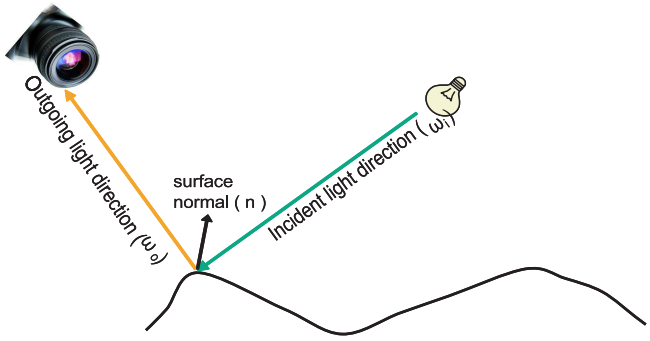


Fig. 2.1 Incident light interacts with surface. The light reflected by the surface enters the camera and is sensed by the image sensor. The sensed intensity depends on several material properties such as (a) surface normal, (b) reflectance characteristics of the surface, and (c) properties of incoming light.

refracted. The amount and direction in which this light is dispersed in each of these modes is dependent on the surface properties. Further, the amount of light reflected along various outgoing directions can vary with the wavelength and direction of incoming light.

For an opaque surface, with no subsurface scattering, the bidirectional reflectance distribution function (BRDF) is a 4D function over the space of all incoming and outgoing directions (2D each) characterizing the ratio of light which is reflected along the outgoing direction to that irradiated on the object along a particular incoming direction. The BRDF is a rich characterization of surface property, which is especially useful in relighting of virtual scenes and is important in many graphics applications. In computer vision applications, where the goal is to estimate and infer properties of the surface (such as surface normals or depth values), simpler models that enable analytical tractability are often used. In particular, the Lambertian model for surface reflectance has found tremendous applications in several vision applications because of its simplicity, analytical tractability, and the range of real-world surfaces that are almost Lambertian in their reflectance. As an example, consider Figure 2.2. The teddy bear doll shown on the left has Lambertian reflectance. This means that the irradiance of outgoing light appears the same in all directions (shown using orange outgoing rays). Many real-world materials have non-Lambertian

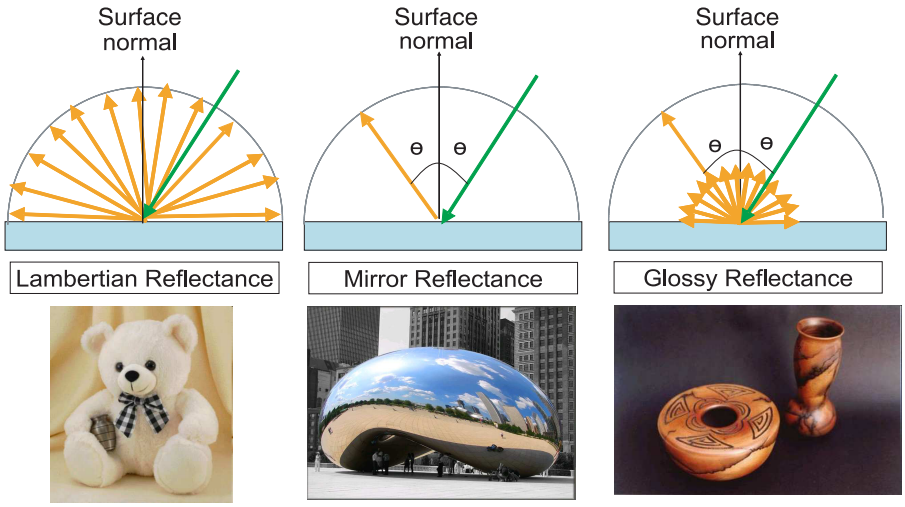


Fig. 2.2 Figure illustrating the reflectance properties of (left) Lambertian, (middle) mirror, and (right) glossy surfaces.

reflectance. The most common example is mirror-like surfaces as shown in Figure 2.2 (middle). Such surfaces reflect incoming light in a specific direction about the local surface normal at the point of incidence. Most real-world surfaces can be adequately characterized using a model that combines the specular and Lambertian components as shown in Figure 2.2 (right). Below, we describe two analytical models for surface reflectance — the Lambertian model and the Phong model.

2.1.1 Lambertian Model

The Lambertian model [123] describes surfaces whose reflectance is independent of the observer's viewing direction. Materials such as matte paint, unpolished wood, and wool exhibit the Lambertian model to a reasonable accuracy. Given a point source of wavelength λ and intensity $I(\lambda)$ illuminating a surface with normal $\mathbf{n}$ at an incident angle $\mathbf{i}$, the reflected light has intensity $I_o(\lambda)$ given as:

$$I_o(\lambda) = \rho(\lambda) I(\lambda) \mathbf{i}^T \mathbf{n}, \tag{2.1}$$

where ρ is the *albedo*, the fraction of energy that is reflected by the surface. In the Lambertian model, the reflectance depends mainly on the

angle between the surface normal and the incident light, encoded in the term $\mathbf{i}^T\mathbf{n}$. The reflected light, or outgoing radiance, is uniform in all outgoing directions, and hence is independent of the observer's viewpoint. For many vision applications, this forms a reasonable approximation of the overall reflectance of the surface. This reflection model is isotropic over the observer's angle of view and does not capture specularities, which are inherently anisotropic. This is illustrated in Figure 2.2 (left).

2.1.2 Phong Model

The Phong model [155] extends the simpler Lambertian model to include specular highlights. As before, given the point light source with intensity I, with incident direction $\mathbf{i}$, the reflected component of this source is given as,

$$I_o(\lambda) = \rho_d(\lambda)I(\lambda)\mathbf{i}^T\mathbf{n} + \rho_s(\lambda)I(\lambda)\left(\mathbf{r}^T\mathbf{v}\right)^\alpha, \tag{2.2}$$

where ρ_d and ρ_s are coefficients describing the ratio of energy in the diffuse and specular components, respectively. $\mathbf{v}$ is the observer's direction (or the outgoing direction of light) and $\mathbf{r} \propto 2(\mathbf{i}^T\mathbf{n})\mathbf{n} - \mathbf{i}$ is the *reflection* direction. α is a constant that controls the nature of the specularity. A high value of α produces a highly localized specularity (like a mirror), while lower values of α produce specularity that is more *even* (such as greasy surfaces). Shown in Figure 2.3 is the rendering of a glossy surface using the Phong illumination model. Note that the ambient contribution is completely flat, while the diffuse contribution accounts for Lambertian shading due to illuminant direction, and the specular

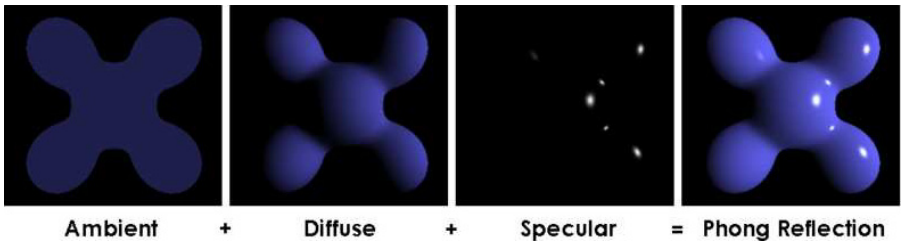


Fig. 2.3 Realistic modeling of surface reflectance can be obtained with the Phong Model. The Phong model encompasses the diffused component (of the Lambertian model) along with specularity (figure courtesy: *Wikipedia*.)

highlights are accounted using the specular term of the Phong model. The resulting rendering is realistic, showing that the Phong reflectance model is an adequate model for several real surfaces.

2.2 Camera Models

Light, after interacting with surfaces, enters the camera and is imaged on the sensor. Lens-based cameras can at most times be reasonably approximated by pinhole cameras (especially when all objects are within its depth of field). The effect of a pinhole camera on light emanating from scene points can be studied through projective geometry. An in-depth discussion of projective geometry can be found in [89, 137]. The projective nature of imaging introduces unique challenges in computer vision, some of which we shall review here.

In the rest of this section, we use **bold-font** to denote vectors and CAPS to denote matrices. Further, we use x, y, z alphabets to denote quantities in world coordinates, and u, v alphabets for image plane coordinates. In addition to this, the concept of homogeneous coordinates is important. We use the *tilde* notation to represent entities in homogeneous coordinates. Given a d-dimensional vector $\mathbf{u} \in \mathbb{R}^{\mathbf{d}}$, its homogeneous representation is given as a $(d+1)$-dimensional vector $\tilde{\mathbf{u}} \backsim [\mathbf{u}, 1]^T$, where the operator $\backsim$ denotes equality up to scale. In other words, $\tilde{\mathbf{u}} \backsim \tilde{\mathbf{x}} \leftrightarrow \tilde{\mathbf{u}} = \lambda\tilde{\mathbf{x}}, \lambda \neq 0$. In simpler terms, when we deal with homogeneous quantities, the scale ambiguity allows for elegant representations of the basic imaging equations, which we discuss next. In particular, the homogeneous representation allows us to write perspective projection as a linear operation.

2.2.1 Central Projection

Central projection is the fundamental principle behind imaging with a pinhole camera, and it serves as a good approximation for lens-based imaging for the applications considered here. In the pinhole camera model, rays (or photons) from the scene are projected onto a planar screen after passing through a pinhole, as illustrated in Figure 2.4. The screen is typically called the image plane of the camera. Consider a camera with its pinhole at the origin and the image plane aligned with

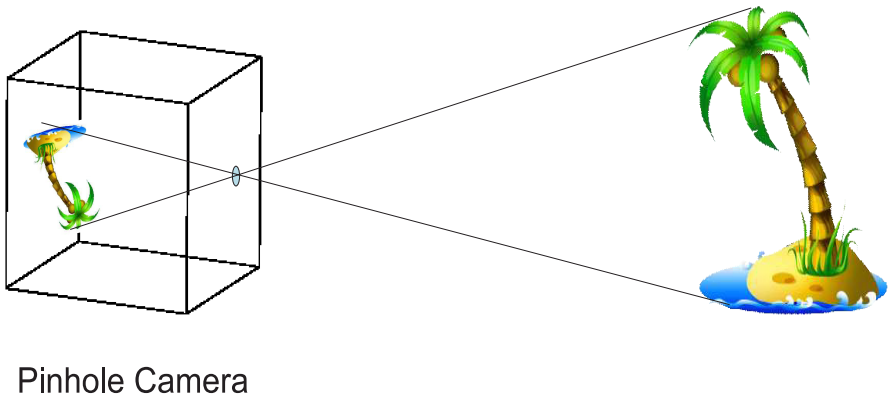


Fig. 2.4 Illustration of the pinhole camera model and central projection. Light from the scene enters the camera through a pinhole and is then sensed on a planar image sensor which is at a focal length f away from the pinhole. The image sensed on the sensor is inverted as shown. Further, the focal length f or the distance between the pinhole and the sensor plane determines the optical magnification.

the plane $z = f$. Under this setup, a 3D point $\mathbf{x} = (x, y, z)^T$ projects onto the image plane point $\mathbf{u} = (u, v)^T$, such that:

$$u = f\frac{x}{z}, \quad v = f\frac{y}{z}. \tag{2.3}$$

This can be elegantly written in homogeneous terms as,

$$\tilde{\mathbf{u}} \backsim \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} fx/z \\ fy/z \\ 1 \end{pmatrix} \backsim \begin{pmatrix} fx \\ fy \\ z \end{pmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}. \tag{2.4}$$

A more general model of the pinhole camera allows for the pinhole to be at an arbitrary position and the image plane oriented arbitrarily. However, we can use a simple Euclidean coordinate transformation to map this as an instance of the previous one. Finally, the camera might have non-square pixels with image plane skew. This leads us to a general camera model whose basic imaging equation is given as:

$$\tilde{\mathbf{u}} \backsim K[R\ \mathbf{t}]\tilde{\mathbf{x}} = P\tilde{\mathbf{x}}, \tag{2.5}$$

where P is the 3×4 matrix encoding both the internal parameters of the camera K (its focal length, principal point, etc.) and the external parameters (its orientation R and position $\mathbf{t}$ in a world coordinate

system). $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{x}}$ are the homogeneous coordinate representations of the pixel in the image plane and the point being imaged in the real world, respectively. Although central projection is inherently non-linear, it can be written as a linear transformation of the homogeneous coordinates. Finally, Equation (2.4) can be obtained from Equation (2.5) with $R = \mathbb{I}_3$ (the identity matrix), $\mathbf{t} = \mathbf{0}$ and $K = \text{diag(f,f,1)}$.

It is noteworthy that the projection equation of Equation (2.5) is not invertible in general. Intuitively, the pinhole camera maps the 3D world onto a 2D plane, so, the mapping is many-to-one and non-invertible. All points that lie on a line passing through the pinhole map onto the same image plane point. This can also be independently verified by the scale ambiguity in Equation (2.5). Given a point on the image plane $\mathbf{u}$, its *pre-image* is defined as the set of all scene points that map onto $\mathbf{u}$ under central projection. It is easily seen that the pre-image of a point is a line in the real world. Without additional knowledge of the scene and/or additional constraints, it is not possible to identify the scene point which projects onto $\mathbf{u}$. The non-invertibility of the scene point (or associated metrics) is a manifestation of the identifiability problem in estimation, mapping more than one (sometimes, infinitely many) point in the solution space. This lack of invertibility leads to some of the classical problems in computer vision, the most fundamental being the establishment of correspondence across views.

2.2.2 Epipolar Geometry

Consider two images (or central projections) of a 3D scene. Given a point $\mathbf{u}_A$ on the first image of a world point $\mathbf{x}$, we know that its pre-image is a line passing through the point $\mathbf{u}_A$ and C_A, the pinhole of the camera (see Figure 2.5). Given information about $\mathbf{u}_A$ on the first image, we can only establish that the corresponding projection of the point $\mathbf{x}$ on the second image plane $\mathbf{u}_B$ lies on the projection of the pre-image of $\mathbf{u}_A$ onto the second image plane. Since the pre-image of $\mathbf{u}_A$ is a line, the projection of this line onto view B gives the line $L(\mathbf{u}_A)$, the *epipolar line* associated with $\mathbf{u}_A$. Thus, epipolar geometry constrains corresponding points to lie on conjugate pairs of epipolar lines.

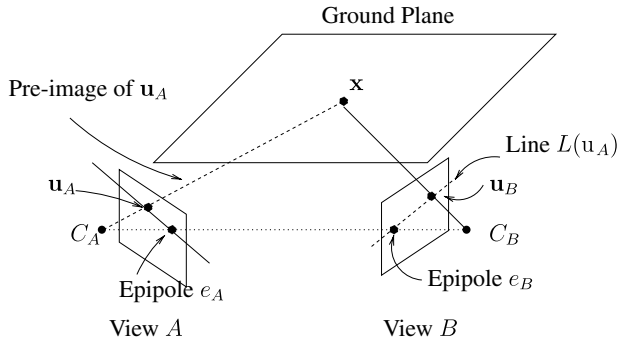


Fig. 2.5 Consider views A and B (camera centers C_A and C_B) of a scene with a point $\mathbf{x}$ imaged as $\mathbf{u}_A$ and $\mathbf{u}_B$ on the two views. Without any additional assumptions, given $\mathbf{u}_A$, we can only constrain $\mathbf{u}_B$ to lie along the image of the pre-image of $\mathbf{u}_A$ (a line). However, if the world were planar (and we knew the relevant calibration information) then we could uniquely invert $\mathbf{u}_A$ to obtain $\mathbf{x}$, and re-project $\mathbf{x}$ to obtain $\mathbf{u}_B$.

Algebraically, the epipolar constraint can be written as a bilinear constraint on the corresponding points $\mathbf{u}_A$ and $\mathbf{u}_B$. A given 3D point (written as $\mathbf{X}_A$ and $\mathbf{X}_B$ in camera-centric coordinate systems) satisfies the Euclidean transformation constraint:

$$\mathbf{X}_B = R\mathbf{X}_A + \mathbf{t}. \tag{2.6}$$

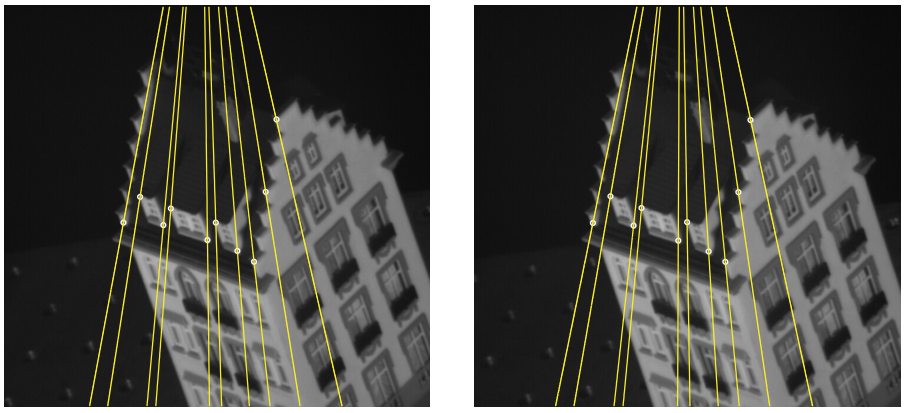
Under projective imaging, the relations $\tilde{\mathbf{u}}_A \backsim K_A X_A$ and $\tilde{\mathbf{u}}_B \backsim K_B X_B$ can be used to rewrite Equation (2.6) as:

$$\begin{aligned} K_B^{-1}\tilde{\mathbf{u}}_B &\backsim RK_A^{-1}\tilde{\mathbf{u}}_A + \mathbf{t}, \\ \mathbf{t}_\times K_B^{-1}\tilde{\mathbf{u}}_B &\backsim \mathbf{t}_\times RK_A^{-1}\tilde{\mathbf{u}}_A + \mathbf{t}_\times\mathbf{t}, \end{aligned} \tag{2.7}$$

where $\mathbf{t}_\times$ is the vector cross product with the vector $\mathbf{t}$. Finally, taking inner products with $K_B^{-1}\tilde{\mathbf{u}}_B$, and noting that $\mathbf{t}_\times\mathbf{t} = \mathbf{0}$, we obtain:

$$\begin{aligned} \tilde{\mathbf{u}}_B^T K_B^{-T}\mathbf{t}_\times K_B^{-1}\tilde{\mathbf{u}}_B &\backsim \tilde{\mathbf{u}}_B^T K_B^{-T}\mathbf{t}_\times RK_A^{-1}\tilde{\mathbf{u}}_A, \\ 0 &\backsim \tilde{\mathbf{u}}_B^T E\tilde{\mathbf{u}}_A, \text{ with } E = K_B^{-T}\mathbf{t}_\times RK_A^{-1}, \end{aligned} \tag{2.8}$$

where $K_B^{-T} = (K_B^{-1})^T$. The matrix E is called the *fundamental matrix* from view A onto view B, and it encodes the epipolar constraint compactly. Given a point $\mathbf{u}_A$, its corresponding epipolar line in view B is $L_B(\mathbf{u}_A) = E\tilde{\mathbf{u}}_A$. Equivalently, the epipolar line in view A corresponding to a point $\mathbf{u}_B$ in view B is given as $L_A(\mathbf{u}_B) = E^T\tilde{\mathbf{u}}_B$.



(a) (b)

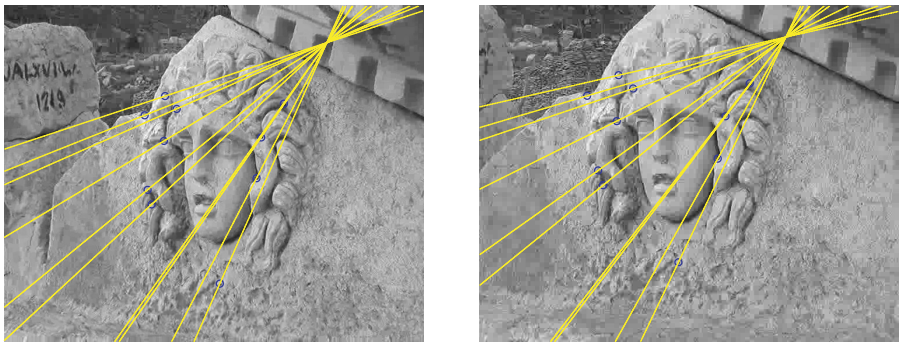
Fig. 2.6 Figure showing two views of a hotel. Feature points are shown in small circles. For each feature point and its corresponding point on the other image, we show epipolar lines. The point where the epipolar lines converge is the epipole. In this case, the epipole is located far from the image center.

We show a few real examples that demonstrate the concepts of epipolar geometry. In Figure 2.6, we show two images from different views of the hotel sequence. Feature points in the images are shown in small circles. Epipolar lines through the corresponding feature points are shown on the second image, and vice versa. We see that the epipolar lines converge outside the image. As another example, consider the Medusa sequence in Figure 2.7, where the epipolar lines converge on the image plane.

In the context of multi-view localization problems, the epipolar constraint can be used to associate objects across multiple views [161]. Once we obtain reliable correspondences across multiple views, we can triangulate to localize objects in the real world. However, correspondences based on the epipolar constraint alone tend to be insufficient, as the constraint does not map points uniquely across views.

2.2.3 Triangulation

In many applications, once we establish correspondences between object locations across views, we are interested in localization of these objects in world coordinates. Let us assume that the same object has



(a) (b)

Fig. 2.7 Figure showing two views of a Medusa head. Feature points are shown in small circles. For each feature point and its corresponding point on the other image, we show epipolar lines. The point where the epipolar lines converge is the epipole.

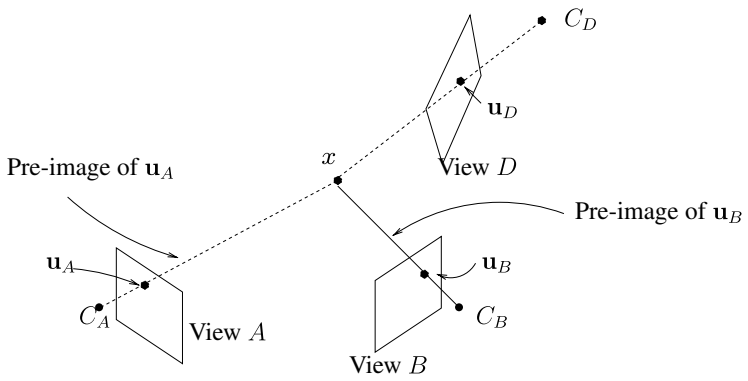


Fig. 2.8 Consider views A, B, and D of a scene with a point $\mathbf{x}$ imaged as $\mathbf{u}_A$, $\mathbf{u}_B$, and $\mathbf{u}_D$ on the views. We can estimate the location of $\mathbf{x}$ by *triangulating* the image plane points as shown in the figure. At each view, we draw the pre-image of the point, which is the line joining the image plane point and the associated camera center. The properties of projective imaging ensure that the world point $\mathbf{x}$ lies on this pre-image. Hence, when we have multiple pre-images (one from each view), the intersection of these lines gives the point $\mathbf{x}$.

been detected in two views (A and B) with camera center C_A and C_B at image plane locations $\mathbf{u}_A$ and $\mathbf{u}_B$ as shown in Figure 2.8. In this case, the basics of projective imaging constrains the object to lie on the pre-image of the point $\mathbf{u}_A$ (the line connecting C_A and $\mathbf{u}_A$). Similarly the object must also lie on the pre-image of $\mathbf{u}_B$ in view B. Therefore, by

estimating the point of intersection of these two lines, we can estimate the true location of the object. In a general scenario with several cameras, each camera generates a line in 3D. By computing the intersection of these lines, the object's location can be estimated. In the presence of noisy measurements, these lines do not intersect at a single point, and error measures such as sum-of-squares are used to obtain a robust estimate of the location of the object. This is called *triangulation* [90]. The drawback of triangulation is that it requires correspondence information across cameras (i.e., which object in camera A corresponds to which object in camera B), which is difficult to obtain.

2.2.4 Planar Scenes and Homography

In certain special scenarios, the imaging equation becomes invertible. An example of such a scenario is when the scene being observed is planar. Most urban scenarios form a good fit as majority of the actions in the world occur in the ground plane. This makes it a reasonable assumption for a host of visual sensing applications. The invertibility can also be efficiently exploited by algorithms for various purposes. As an example, consider the correspondence problem we mentioned earlier. Under a planar world assumption, the pre-image of a point becomes a point (in most cases), being the intersection of the world plane and the pre-image line. This implies that by projecting this world point back onto the second image plane, we can easily find correspondence between points on the two image planes. This property induced by the world plane allows for finding correspondences across image planes, and is referred to as the *homography* induced by the plane.

Consider two views of a planar scene labeled view A and view B. We can define a local coordinate system at each view. The same scene point is represented as $\mathbf{x}_A$ and $\mathbf{x}_B$ on the two coordinate systems, and they are related by the Euclidean transformation,

$$\mathbf{x}_B = R\mathbf{x}_A + \mathbf{t}. \tag{2.9}$$

Here, R (a rotation matrix) and $\mathbf{t}$ (a 3D translation vector) define the coordinate transformation from A to B. Let us assume that the world

plane has an equation $\mathbf{n}^T\mathbf{x}_A = d$ with $d \neq 0$.[1] For points that lie on the plane, we can rewrite Equation (2.9) as,

$$\begin{aligned} \mathbf{x}_B &= R\mathbf{x}_A + \mathbf{t}\frac{\mathbf{n}^T\mathbf{x}_A}{d} \\ &= \left(R + \tfrac{1}{d}\mathbf{t}\mathbf{n}^T\right)\mathbf{x}_A. \end{aligned} \tag{2.10}$$

In each local camera coordinate system, we know that $\tilde{\mathbf{u}} \backsim K[R\ \mathbf{t}]\tilde{\mathbf{x}}$ (see Equation (2.5)) with $R = \mathbb{I}_3$ and $\mathbf{t} = \mathbf{0}$. Therefore, $\tilde{\mathbf{u}}_B \backsim K_B\mathbf{x}_B$ and $\tilde{\mathbf{u}}_A \backsim K_A\mathbf{x}_A$, which gives us,

$$\begin{aligned} K_B^{-1}\tilde{\mathbf{u}}_B &\backsim \left(R + \tfrac{1}{d}\mathbf{t}\mathbf{n}^T\right)K_A^{-1}\tilde{\mathbf{u}}_A, \\ \tilde{\mathbf{u}}_B &\backsim H\tilde{\mathbf{u}}_A, \text{ where } H = K_B\left(R + \tfrac{1}{d}\mathbf{t}\mathbf{n}^T\right)K_A^{-1}. \end{aligned} \tag{2.11}$$

This implies that a point in view A, $\mathbf{u}_A$ maps to the point $\mathbf{u}_B$ in view B as defined by the relationship in Equation (2.11). The 3×3 matrix H (in Equation (2.11)) is called the homography matrix, or simply, the homography. Also, note that H (like P) is a homogeneous matrix, and the transformation it defines is unchanged when H is scaled. Further, H is invertible when the world plane does not pass through pinholes at either of the two views. This is easily verified as our derivation is symmetric in its assumptions regarding the two views.

Finally, the premise of the induced homography critically depends on the fact that the pre-image of a point on the image plane is a unique point on the world plane. Suppose we use a local 2D coordinate system over the world plane, the image plane to world plane transformation (from their respectively 2D coordinate systems) can be shown to be a projective transformation, which can be encoded as a 3×3 homogeneous matrix, say H_π. This transformation is useful when we want to estimate metric quantities, or quantities in an Euclidean setting. The most common example of this happens when we need to localize the target in the scene coordinates.

Computing the image plane to world plane transformation H_π presents a challenging problem, which is typically accomplished by exploiting properties of parallel and perpendicular lines on the planes. Typically, this requires manual inputs such as identifying straight line

[1] When $d = 0$, the plane passes through the pinhole at A, thereby making the imaging non-invertible

segments that are parallel. While not always possible, many urban scenes (such as parking lots, roads, buildings) contain such lines that make it easier to estimate the transformation H_π, at least in a semi-supervised manner. Computing H_π, as it happens, is identical to a metric rectification of the image plane. Many such techniques are illustrated in [89].

2.2.5 A Note on Calibration

In order to use the epipolar constraint or the homography constraint on images obtained from arbitrary camera views, it is essential to first calibrate the various cameras. Calibration of a camera refers to the estimation of camera parameters, such as its internal parameters K or its external parameters in terms of rotation R and translation $\mathbf{t}$ with respect to world coordinates. However, in some cases, it might suffice to estimate a subset or a minimal encoding of these parameters. The two-view problem exemplifies this, where the fundamental (or essential) matrix completely encodes the pertinent geometric information. Under the plane-based homography constraint, calibration could involve estimating the 3×3 homography matrix, H, which encodes the relation between the image location and the corresponding location on the scene plane. In either case, calibration is an extremely well-studied problem, and we refer the readers to [89, 137] for an in-depth analysis of the methods and issues involved in calibration. In particular, we highlight that geometric constructs, such as rotation matrices, fundamental matrices, and camera internal parameters lie on special matrix manifolds. Exploiting their special structure leads to efficient and robust solutions.

2.2.6 Simplifications to the Perspective Imaging Model

All the properties discussed so far, including epipolar geometry and homography, are artifacts introduced due to central projection (the model of imaging with an ideal pinhole), which serves as a good approximation to lens-based imaging. Note that the mapping of 3D points onto the image plane is non-linear under this imaging model. This is also accompanied by some loss of information (a 3D to 2D mapping).

However, for many scenarios, alternate models can serve as approximations to the central projection model providing analytic tractability [65]. We discuss two such models here: the *weak perspective* and the *para-perspective* models.

The weak perspective (or scaled orthography) model assumes that the object's depth variations are negligible, so, the imaging equation can be approximated by using a reference depth z_0. With this, the depth of a scene point z be approximated as $z = z_0(1+\epsilon)$ where $\epsilon = 1 - z/z_0$ is minute or minimal under the assumptions. With this, the imaging equations can be written as,

$$u = f\frac{x}{z} = f\frac{x}{z_0(1+\epsilon)} \approx f\frac{x}{z_0}, \tag{2.12}$$

$$v = f\frac{y}{z} = f\frac{y}{z_0(1+\epsilon)} \approx f\frac{y}{z_0}. \tag{2.13}$$

However, the error in the approximation in the weak perspective model becomes large, especially for large fields of view. In the para-perspective model [91], a first-order approximation of the perspective imaging model is used to obtain a better approximation as follows:

$$u = f\frac{x}{z} = f\frac{x}{z_0(1+\epsilon)} \approx f\frac{x}{z_0}(1-\epsilon), \tag{2.14}$$

$$v = f\frac{y}{z} = f\frac{y}{z_0(1+\epsilon)} \approx f\frac{y}{z_0}(1-\epsilon). \tag{2.15}$$

The relation in Equation (2.15) is the para-perspective imaging model. It is bilinear in the world coordinates (x, y, z), which are often easier to handle than the non-linearity of the central projection.

2.3 Motion

Camera motion and object motion lead to several interesting vision applications. Camera motion leads to observability of certain quantities (such as an object's structure) that are typically unobservable in the static single camera scenario. Further, in many applications such as surveillance, moving objects are of primary interests to us. It is important to study motion of objects both at a macroscopic level (the level of the object and its parts) and a microscopic level (the level of point

features on the object). This results in a range of models for motion that arise in various contexts. We discuss a few of these motion models here.

2.3.1 Microscopic Model: Brightness Constancy and Optical Flow

Estimating motion for point features is referred to as the problem of estimating the *optical flow*. Optical flow is defined as the apparent flow induced on the image plane due to the real motion of the object [92]. Typically, we need to make additional assumptions to solve this problem. We discuss the *brightness constancy* assumption. This assumes that the brightness of a point feature on the object remains the same even when the object is moving, or the camera is moving.

Let $I(x, y, t)$ be the intensity observed at pixel (x, y) at time t. The principle of brightness constancy suggests that for a point object that undergoes translation from a point (x, y) at time t to $(x + \Delta x, y + \Delta y)$ at time $t + \Delta t$, the observed intensity remains the same, i.e.,

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t). \tag{2.16}$$

Using the Taylor series expansion we obtain,

$$I(x + \Delta x, y + \Delta y, t + \Delta t) \approx I(x, y, t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t. \tag{2.17}$$

Substituting this expression in Equation (2.16),

$$\begin{aligned} \tfrac{\partial I}{\partial x}\Delta x + \tfrac{\partial I}{\partial y}\Delta y + \tfrac{\partial I}{\partial t}\Delta t &= 0, \\ \tfrac{\partial I}{\partial x}\tfrac{\Delta x}{\Delta t} + \tfrac{\partial I}{\partial y}\tfrac{\Delta y}{\Delta t} &= -\tfrac{\partial I}{\partial t}, \\ \nabla I \cdot \mathbf{v} &= -I_t, \end{aligned} \tag{2.18}$$

where $\mathbf{v}$ is the apparent velocity of the point on the image plane, in other words, the flow vector, $\nabla I = (\partial I/\partial x, \partial I/\partial y)$, denotes the spatial gradients of the image in x and y directions, and I_t denotes the temporal gradient of the image sequence. The expression $\nabla I \cdot \mathbf{v} = -I_t$ expresses a fundamental relationship relating the spatial and temporal gradients to the flow vector. Note that, the flow vector $\mathbf{v}$ has two components and the optical flow equation (2.18) provides only one equation.

The Lucas–Kanade algorithm [202] solves for the flow by assuming a constant flow for all points in a small neighborhood of the feature point. Each point now provides a constraint in $\mathbf{v}$, and by grouping all such constraints, and solving for a minimum mean-square error (MMSE) solution, an estimate for the flow can be obtained.

$$\hat{\mathbf{v}} = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{bmatrix} = \mathbf{A}^{-1}\mathbf{b}, \tag{2.19}$$

where the summation is for all pixels in a neighborhood, and I_x and I_y denote $\partial I/\partial x$ and $\partial I/\partial y$, respectively.

A robust solution for the flow vector using Equation (2.19) is possible only when the matrix $\mathbf{A}$ is well-conditioned. Typically, for smooth regions in the image the matrix has a rank 0, and for points along single edges, it has rank 1. This is referred to as the *aperture problem*, and it is a fundamental limitation of flow estimation using local information. It is only for neighborhoods that have strong gradients in multiple directions (such as corners and intersections), that the matrix is well conditioned and invertible. In Figure 2.9, two images of the same scene captured from a moving camera are shown. On the right, the computed dense optical flow field is shown with red arrows indicating direction and magnitude of apparent motion.

Alternate formulations exist for modeling the optical flow field, along with a variety of estimation algorithms to solve for the flow. While optical flow is a description of the flow field for a point and

Fig. 2.9 (left) Two images from the Middlebury data set [8], (right) and the optical flow capturing dense motion between the two images.

its neighborhood, more sophisticated models are used to describe the macroscopic motion of objects.

2.3.2 Macroscopic Motion Models: Rigid, Brownian and Constant Velocity Motion

Rigid body motion, or equivalently *Euclidean motion*, encompasses rotation and translation of an object. A 3D point X on an object under rigid motion obeys the following motion model,

$$\mathbf{X}(t) = R_t \mathbf{X}(0) + \mathbf{T}_t, \tag{2.20}$$

where $R_t \in SO(3)$ is the 3×3 matrix defining the rotation[2] at time t, and $\mathbf{T}_t$ is the 3D translation vector at time t. Rigid motion preserves angles and distances on the 3D body as shown in Figure 2.10. However, under the perspective imaging model, the apparent motion produced by the object on the image plane is no longer Euclidean, and *without the knowledge of the 3D structure of the object*, the motion has little symmetry to it. To address this problem, a set of image plane motion

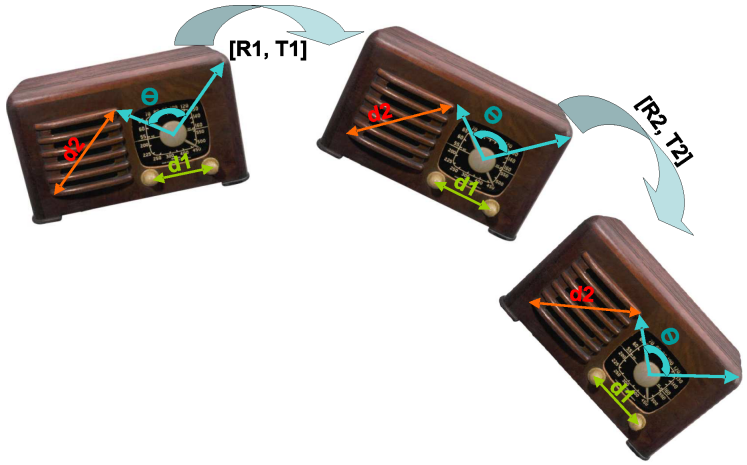


Fig. 2.10 Rigid body motion preserves angles and distances.

[2] The properties of rotation matrices, along with their various parameterizations, are of immense use in vision. The interested reader is directed to [197] for an in-depth discussion of rotation matrices.

models exist that approximate the entire motion of the object over short time segments.

Brownian motion is the simplest of image plane motion models. Let θ_t be a description of the object state (such as its location on the image plane, its shape). The temporal evolution of this state given as,

$$\theta_t = \theta_{t-1} + \omega_t, \tag{2.21}$$

where ω_t is a zero-mean noise process (typically Gaussian). The property of such a motion model is that $\mathrm{E}(\theta_t) = \mathrm{E}(\theta_{t-1})$ and $\mathrm{var}(\theta_t) \geq \mathrm{var}(\theta_{t-1})$. On an average, trajectories from this model tend to concentrate around its initial value, however, the variations about the mean state increase with time. The Brownian motion model is generic, and can accommodate a wide range of motion trajectories, making it a useful model for situations where no apparent symmetry occurs in the observed motion.

The constant velocity model assumes that the velocity of the state vector remains constant over short durations of time, leading to the following state transition model,

$$\theta_t = \theta_{t-1} + \dot{\theta}_{t-1} + \omega_{t,1}, \tag{2.22}$$

$$\dot{\theta}_t = \dot{\theta}_{t-1} + \omega_{t,2}, \tag{2.23}$$

where $\omega_{t,1}, \omega_{t,2}$ are noise processes, and $\dot{\theta}_t$ is the velocity of the state. In many cases, the inertia associated with object movement makes a constant velocity assumption a suitable model, especially when the imaging can be approximated well with an orthographic model. This model possesses additional smoothness in comparison to the Brownian motion model, at the cost of estimation of additional parameters. Models such as these can be extended to model more complicated behaviors including accelerations, and curved trajectories.

3

Statistical Estimation Techniques

In the previous section, we discussed the geometric constraints that arise from various physical processes commonly observed in vision. In this section, we review some basics of statistical estimation and inference that form the basis of the remainder of the discussion. Statistical estimation can be broken down broadly into *static* and *dynamic* estimation. Static estimation refers to parameters fixed in time (or for problems with no concept of time). Dynamic estimation refers to parameters that are time-varying. In particular, we concentrate on time-varying parameters whose characterizations are available in terms of *dynamical systems*. Dynamical systems form powerful models for describing time-varying parameters and are used in a wide range of problems. We will discuss the simple maximum likelihood estimator, the exact MAP estimate, and approximations to the exact solution. Statistical estimation plays an important role in target tracking applications to be discussed in Section 4. We will also discuss how the goodness of estimators is quantified via Cramer–Rao lower bounds. These bounds will play an important role in quantifying the quality of structure from motion algorithms in Section 5.

3.1 Static Estimation

3.1.1 Maximum Likelihood Estimation

Consider a vector of parameters of interest $\theta \in \mathbb{R}^d$, and observation $\mathbf{y} \in \mathbb{R}^p$ that obey the model $p(\mathbf{y}|\theta)$. Let us assume that multiple independent observations $\mathbf{y}_{1:N} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$ are drawn from the same model $p(\cdot|\theta)$. The maximum likelihood estimate (MLE) θ_{MLE} is obtained by maximizing the observation likelihood given the model, i.e.,

$$\theta_{\text{MLE}} = \arg\max_{\theta} p(\mathbf{y}_1, \ldots, \mathbf{y}_N|\theta) = \arg\max_{\theta} \Pi_{i=1}^{N} p(\mathbf{y}_i|\theta). \tag{3.1}$$

The MLE is a useful estimation technique that is easy to obtain in many cases, especially given a simple likelihood model. It also has favorable asymptotic behavior as the number of the observations increases.

3.1.2 Bayesian Estimation

Additional information of θ available in the form of a priori density $p(\theta)$ can also be incorporated into the parameter estimation. This leads to the maximum a posteriori (MAP) estimator, θ_{MAP}, defined as

$$\theta_{\text{MAP}} = \arg\max_{\theta} p(\mathbf{y}|\theta)p(\theta). \tag{3.2}$$

The MAP estimate differs from the MLE because it assumes that the unknown parameter θ is a random variable with a priori characterization $p(\theta)$. In many cases, we are interested in a complete characterization of the uncertainty of the parameter θ over its state space. This is done by estimating the probability density over the parameter space conditioned on the observations. Formally, given a set of independent and identically distributed (i.i.d.) observations $\mathbf{y}_{1:N}$, the goal is to compute the posterior density function $p(\theta|\mathbf{y}_{1:N})$. To estimate this, we can use Bayes' theorem as,

$$\begin{aligned} p(\theta|\mathbf{y}_{1:N}) &= \frac{p(\mathbf{y}_{1:N}|\theta)p(\theta)}{p(\mathbf{y}_{1:N})}, \\ &= \frac{p(\theta)\Pi_{i=1}^{N} p(\mathbf{y}_i|\theta)}{p(\mathbf{y}_{1:N})}. \end{aligned} \tag{3.3}$$

Estimation of the posterior $p(\theta|\mathbf{y}_{1:N})$ is fundamental in *Bayesian inference*. Estimation of the posterior density is analytically possible only for a limited class of problems.

3.1.3 Approximate Methods

As discussed above, estimation of the posterior is in general analytically intractable. There exist many approximate inference methods for estimating the posterior, as discussed below.

Laplace's method [64] approximates the posterior as a Gaussian distribution, whose parameters are obtained by a local linear approximation of the posterior about a pivot point. The pivot point is typically a local mode of the posterior or the MLE, if available. Laplace's method provides a good approximation when the posterior is unimodal.

Graphical models [121, 124] represent the variables (both state and observations) as nodes of a graph, and the edges map the dependencies between the variables. They effectively model complex relationships between variables. The sum–product algorithm solves the inference problem for graphical models when the representation of the graph is in the form of a factor graph. The sum–product algorithm is exact when the factor graph is a tree, and in practice has been observed to provide accurate solutions for a wide range of graphs.

Variational inference approximates the true posterior with another from a family of distributions that are analytically simpler [103]. Given the observations, the parameters of the approximate density are chosen by minimizing the Kullback–Leibler (KL) divergence to the true posterior. A qualitative understanding of the form of the posterior, and the selection of a family of distributions that approximate the posterior accurately, is crucial to variational Bayesian inference.

Conjugate priors assume importance when the likelihood density belongs to a parametric family [49]. When the prior $p(\theta)$ and the likelihood $p(\mathbf{y}|\theta)$ form a conjugate pair, the posterior density $p(\theta|\mathbf{y}_{1:N})$ as defined in Equation (3.3) has the same distribution as the prior, albeit with different parameters. Many conjugate pairs [49] exist that are useful in a wide range of estimation problems. Conjugate priors provide analytical solutions, as well as efficient numerical estimates. Conjugate priors are known for a limited class of distributions such as the exponential family, and in general are not available for complicated density models.

Monte Carlo methods address the estimation of the posterior density by generating samples from the posterior itself. These samples are then used to estimate quantities of interest, such as the mean and the variance of the posterior density function. Sampling from an arbitrary density is, in general, a difficult problem. However, there exist general purpose sampling schemes that generate samples from a proposal or a sampling density (also called an *importance function*) and modify the sample set so it is statistically equivalent to samples from the posterior density. Examples of such methods include accept–reject sampling, importance sampling and the Metropolis–Hastings sampling algorithm. For an in-depth discussion of this topic we refer the reader to existing literature on sampling [141, 168].

Monte Carlo techniques have found use in many computer vision problems. One early example was the computation of MAP estimates in Markov random fields. Geman and Geman [76] used the Gibbs sampler (a special case of the Metropolis–Hastings algorithm) to sample from the posterior of the random field, which is used for image restoration and segmentation. Monte Carlo techniques find extensive use in the Bayesian inference of non-linear non-Gaussian dynamical systems. The use of *particle filters* is now ubiquitous in a wide range of vision applications, including visual tracking [97] and structure from motion [159]. We discuss these in greater detail in Section 3.4.

3.2 Robust M-Estimators

In statistical estimation, estimators need to be robust to outliers to the modeling assumptions. As a simple example, let the modeling assumption be that the observed data arises from a scalar normal distribution with an unknown mean μ and known variance σ^2. Given data $\{x_i, i = 1, \ldots, N\}$ such that $x_i \backsim \mathrm{N}(\mu, \sigma^2)$, the sample mean is defined as:

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} x_i. \tag{3.4}$$

It can be shown that $\hat{\mu}$ is an unbiased estimate of μ ($\mathrm{E}(\hat{\mu}) = \mu$, and $\mathrm{var}(\hat{\mu}) = \sigma^2/N$), and it achieves the Cramer–Rao lower bound (CRLB)

on the variance, making it an optimal estimator. However, consider the case when the data are corrupted with outlier noise. Let the true data be from a mixture of a normal distribution $\mathrm{N}(\mu,\sigma^2)$ and a corrupting distribution f_c, then,

$$x_i \backsim (1-\epsilon)\mathrm{N}(\mu,\sigma^2) + \epsilon f_c,\ 0 < \epsilon < 1. \tag{3.5}$$

If the mean of the corrupting distribution $\mu_c \neq \mu$, we can immediately see that the sample mean would be biased $\mathrm{E}(\hat{\mu}) = (1-\epsilon)\mu + \epsilon\mu_c = \mu + \epsilon(\mu_c - \mu) \neq \mu$. The amount of bias would be proportional to the value of ϵ and the mismatch between the means $(\mu - \mu_c)$. In the worst case, such a bias could be catastrophic if the corrupting distribution was *Cauchy*, since the mean of a Cauchy distribution is undefined. In such a setting, μ_c is undefined, and the variance of the corrupting distribution is infinite. Thus, the expected value of the sample mean $\hat{\mu}$ is undefined, and its variance is infinite. A more general discussion can be found in [173]. So, it is important to design estimators that are *robust* to outliers, especially when the outlier model is unknown. Consider the simple instructive example of estimating the parameters of a line from data with additive Gaussian noise. Further, assume that (as shown in Figure 3.1) there are some outliers (three outliers in the shown example). Notice that the three outliers cause a significant bias in the MMSE estimation of the line. To account for the effect of these outliers, one must construct an estimator that is robust to outliers. M-estimators are a class of such estimators which (as shown in Figure 3.1) are robust to outliers.

Let us now discuss how M-estimators provide robust methods for parameter estimation. We will discuss the problem of mean estimation from sample points to illustrate these ideas. The sample mean is the estimate that minimizes the mean square error of representing a sample set with a single point,

$$\hat{\mu} = \arg\min_{x} (1/N) \sum_{i=1}^{N} \|x - x_i\|_2^2. \tag{3.6}$$

Each sample x_i contributes $\|x_i - \hat{\mu}\|_2^2$ or a quadratic error term to this cost. Due to this, samples that are far away from the solution point

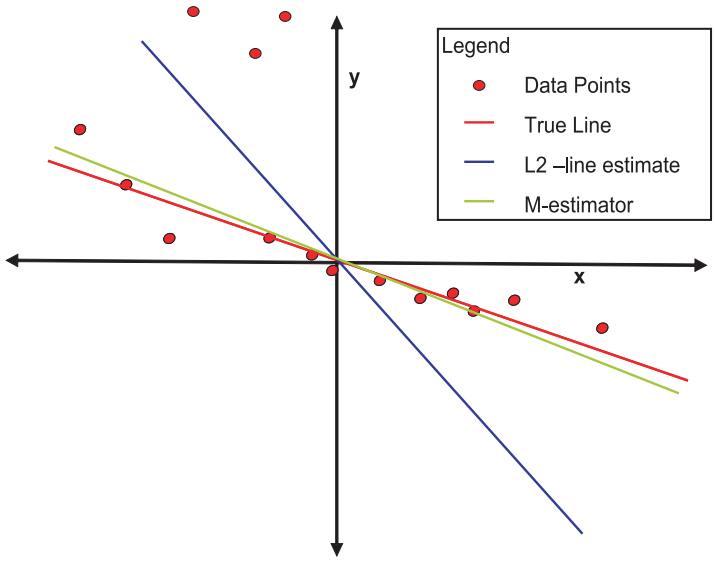


Fig. 3.1 An example illustrating line-fitting in the presence of noisy data and outliers. Least squares fitting causes a large error in the estimated line, whereas robust fitting comes closer to the true line.

$\hat{\mu}$ contribute much more to the resultant error than samples that are close to the solution. This makes the sample mean extremely sensitive to outlier points. Thus, we need to use cost functions that are inherently robust to outliers. An example of such robust statistic is the *median*. The estimator

$$\hat{\mu}_{\text{med}} = \arg\min_{x} \ \text{median}\{\|x - x_i\|_2^2, i = 1, \ldots, N\} \tag{3.7}$$

minimizes the median of the individual error terms (as opposed to their mean), and is called the least median square error (or LMedSE) estimator [170, 140]. The LMedSE is also an unbiased estimator of μ, but does not achieve the CRLB for finite sample sets. As the sample size grows, it asymptotically achieves the lower bound. However, it is far more robust to outlier points.

We have so far seen two estimation criteria: the mean square error and the median square error. *M-estimators* [95, 229] are a generalization of this basic concept to a class of estimators. Let $\psi : \mathbb{R}^d \to \mathbb{R}$ be a

function (called a cost function). Define an estimator $\hat{\mu}_\psi$ as:

$$\hat{\mu}_\psi = \arg\min_x \sum_{i=1}^{N} \psi(x - x_i). \tag{3.8}$$

The choice of ψ allows us to control the nature of the estimator: its convergence properties, and how well it handles inliers and outliers. If $\psi(x) = \|x\|_2^2$, then the resulting estimator is the MMSE. $\psi(x) = \|x\|_1$ produces an estimator that mirrors the median. Many popular choices exist for the function ψ, including the *Huber* and *Tukey* functions. Some typical robust cost functions that are used to fit regression models to observed data are shown in Figure 3.2.

We also direct the interested reader to the random sample consensus (RANSAC) algorithm [68, 140]. RANSAC handles outliers by choosing a subset of the sample points, estimating the parameter of interest based on this subset, then classifying the whole sample set in terms

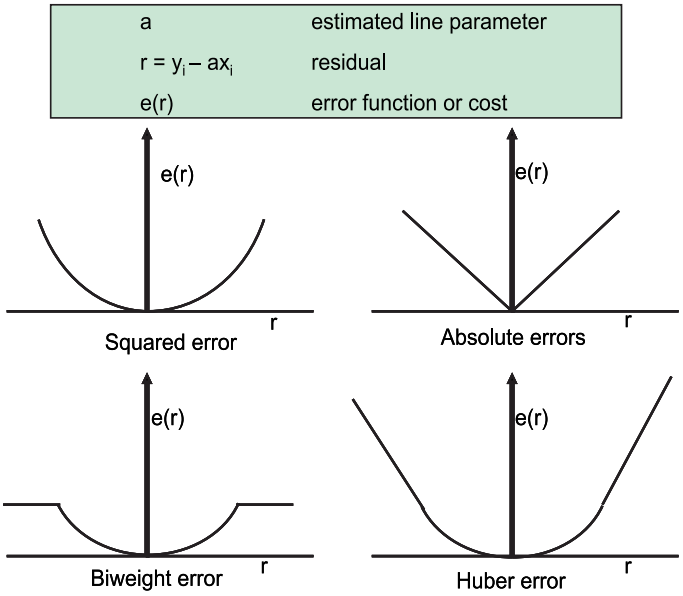


Fig. 3.2 Typical cost functions that are used in order to fit regression models to observed data. The squared error cost function is most sensitive to outliers while the biweight error is most robust to outliers. Huber cost function and the absolute error cost function are less sensitive to outliers than the squared error cost function.

of inliers and outliers. Several repetitions of this process lead to the selection of the subset with maximal number of inliers. The RANSAC algorithm works extremely well even with a high percentage of outliers in data.

3.3 Performance Evaluation of Statistical Methods

The random nature of the measurements implies that any estimator based on them will have an associated uncertainty. One way to quantify the goodness of an unbiased estimator $\hat{\theta}$ is to compute the covariance matrix of the estimation error conditioned on the true parameters θ,

$$E\{(\hat{\theta}-\theta)(\hat{\theta}-\theta)^{\mathbf{T}}|\theta\}, \tag{3.9}$$

where E is the expectation operator. Estimators $\hat{\theta}$ with smaller error covariance are considered better because less uncertainty is involved. Assume that the conditional probability density function of observations $\mathbf{y}$ given by $f(\mathbf{y}|\theta)$ is known. Then, the uncertainty of an unbiased estimator $\hat{\theta}$ is bounded below by the Cramer–Rao inequality [166, 189]:

$$E\{(\hat{\theta}-\theta)(\hat{\theta}-\theta)^{\mathbf{T}}|\theta\} \geq \mathbf{J^{-1}}, \tag{3.10}$$

where J^{-1} is the inverse (assuming it exists) of the Fisher information matrix, given by,

$$J = E\left\{[\frac{\partial}{\partial\theta}\ln f(\mathbf{y}|\theta)]^{\mathbf{T}}[\frac{\partial}{\partial\theta}\ln \mathbf{f}(\mathbf{y}|\theta)]|\theta\right\}, \tag{3.11}$$

and the inequality sign '$\geq$' in Equation (3.10) is defined for symmetric matrices A and B, where $A \geq B$ implies that $(A-B)$ is positive semidefinite. Hence, Equation (3.10) implies that the matrix

$$E\{(\hat{\theta}-\theta)(\hat{\theta}-\theta)^{\mathbf{T}}|\theta\} - \mathbf{J^{-1}}$$

is positive semi-definite. Since all positive semi-definite matrices have non-negative diagonal terms,

$$E\{(\hat{\theta}-\theta)(\hat{\theta}-\theta)^{\mathbf{T}}|\theta\}_{\mathbf{ii}} \geq \mathbf{J_{ii}^{-1}}. \tag{3.12}$$

Let us denote the i-th component of $\hat{\theta}$ by $\hat{\theta}_i$. Since the i-th diagonal term of $E\{(\hat{\theta}-\theta)(\hat{\theta}-\theta)^{\mathbf{T}}|\theta\}$ is the error variance of of $\hat{\theta}_i$, we have:

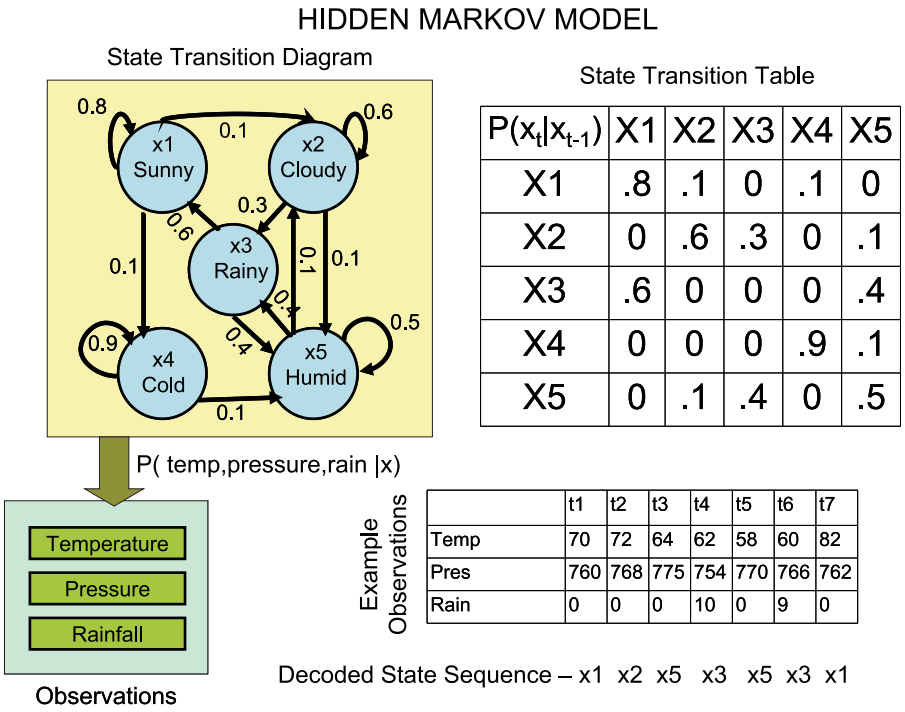
$$\text{error variance of } \hat{\theta}_i \geq \text{the } i\text{-th diagonal term of } J^{-1}(\equiv \text{CRLB of } \theta_i). \tag{3.13}$$

The i-th diagonal term of J^{-1} is called the Cramer–Rao lower bound (CRLB) of θ_i because it lower bounds the error variance of any estimator of θ_i. Since the Fisher information matrix J is independent of $\hat{\theta}$ by Equation (3.11), the CRLBs are independent of the estimate $\hat{\theta}$ by Equation (3.13). No matter which unbiased estimator is applied, the error variances of the estimated parameters can never be reduced below the CRLB. On the other hand, CRLBs depend on the actual parameter θ (J is a function of θ). The dependence of the CRLB on the underlying parameters, and other problem variables, is useful in the study of inherent ambiguities in estimation.

3.4 Dynamical Systems for Estimation

Dynamical systems provide a structured representation for both the nature of the temporal variations, and the relationship between observations and the time-varying parameter. Here, we formulate the problem of Bayesian inference for dynamical systems. Let $\mathcal{X}$ and $\mathcal{Y}$ denote the state space and the observation space of the system, respectively. Let $x_t \in \mathcal{X}$ denote the state at time t, and $y_t \in \mathcal{Y}$ the noisy observation at time t. We model the state sequence $\{x_t\}$ as a Markovian random process. Further we assume that the observations $\{y_t\}$ to be conditionally independent given the state sequence. Under these assumptions, the system is completely characterized by the following:

- $p(x_t|x_{t-1})$: The *state transition density*, describing the evolution of the system from time $t-1$ to t. Alternatively, the same could be described with a *state transition model* of the form $x_t = h(x_{t-1}, n_t)$, where n_t is a noise process. Figure 3.3 gives an example of a state transition matrix for a discrete state space.
- $p(y_t|x_t)$: The *observation likelihood density*, describing the conditional likelihood of observations given the state. As before, this relationship could take the form of an *observation model* $y_t = f(x_t, \omega_t)$ where ω_t is a noise process independent of n_t. Notice that the temperature, pressure, and rainfall



$P(x_t\|x_{t-1})$	X1	X2	X3	X4	X5
X1	.8	.1	0	.1	0
X2	0	.6	.3	0	.1
X3	.6	0	0	0	.4
X4	0	0	0	.9	.1
X5	0	.1	.4	0	.5

	t1	t2	t3	t4	t5	t6	t7
Temp	70	72	64	62	58	60	82
Pres	760	768	775	754	770	766	762
Rain	0	0	0	10	0	9	0

Fig. 3.3 Example of a hidden Markov model (HMM) for 'weather'. The state transition diagram is a finite state machine, while the temperature, pressure, and rainfall observations are recorded. The state transition table and the conditional observation model can be used to determine the most likely state sequence using Viterbi decoding.

observations obtained in Figure 3.3 are conditionally dependent on the state (weather).

- $p(x_0)$: The prior state probability at $t = 0$.

Given statistical descriptions of the models and noisy observations, we are interested in inferencing the state of the system at the current time. Specifically, given the observations till time $t, y_{1:t} = \{y_1, \ldots, y_t\}$, we would like to estimate the posterior density function $\pi_t = p(x_t|y_{1:t})$. With the posterior, we aim to make inferences $I(f_t)$ of the form,

$$I(f_t) = \mathbf{E}_{\pi_t}[f_t(x_t)] = \int f_t(x_t)p(x_t|y_{1:t})dx_t, \tag{3.14}$$

where f_t is some function of interest. An example of such an inference is the conditional mean, where $f_t(x_t) = x_t$. Under the Markovian

assumption on the state space dynamics and the conditional independence assumption on the observation model, the posterior probability π_t is recursively estimated using the *Bayes Theorem* as:

$$\pi_t = p(x_t|y_{1:t}) = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})} = \frac{p(y_t|x_t)p(x_t|y_{1:t-1})}{\int p(y_t|x_t)p(x_t|y_{1:t-1})dx_{t-1}}. \tag{3.15}$$

The computation of $p(x_t|y_{1:t-1})$ sets up the premise for recursion and is called the *prediction* step,

$$p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1}. \tag{3.16}$$

Note that

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t)\int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1}}{p(y_t|y_{1:t-1})} \tag{3.17}$$

has no unknowns because all terms are either specified or computable from the posterior at the previous time step. The problem is that this computation (including the integrations) need not have an analytical representation. Analytical solutions exist when the state transition model and the observation model are both linear, and we are interested in tracking only the first two moments (the mean and the covariance matrix) of the posterior density function. In this setting, the optimal estimator is the *Kalman filter* [107]. For non-Gaussian noise models, the Kalman filter is optimal among the class of linear filters.

In the more general case when we are interested in higher order statistics, the inference problem becomes harder. The general inference problem can be efficiently solved for two cases. The first is, when the state space $\mathcal{X}$ is a finite discrete set, the dynamical system can be compactly modeled as a finite state machine, and the inference problem can be solved efficiently as well. The second case is when we allow for approximate inference, leading to a class of sequential Monte Carlo techniques also known as particle filters [56, 80, 130].

3.4.1 Finite State Machines: Hidden Markov Models

When the state space $\mathcal{X}$ is a finite set $\{1, \ldots, M\}$, we can compactly represent the state transition model in terms of an $M \times M$ matrix.

Suppose we define $\Pr(x_t = j | x_{t-1} = i) = p_{ij}^t$, such that $0 < p_{ij}^t < 1$. The matrix $P(t) = [p_{ij}^t]$ is the $M \times M$ state transition matrix, and must satisfy

$$\sum_j p_{ij}^t = 1 \iff P(t)\mathbf{1} = \mathbf{1}, \tag{3.18}$$

implying that the all-one vector is one of its eigenvectors with unit eigenvalue. An example of a state-transition matrix for weather prediction is shown in Figure 3.3.

The posterior probability mass function[1] π_t such that $\pi_t^T \mathbf{1} = 1$ and each element of π_t is a number between 0 and 1 can be recursively estimated. The vector denoting the probability mass of $Pr(x_t | y_{1:t-1})$ is given as $P(t)^T \pi_{t-1}$. Let $L_y \in \mathbb{R}^M$ be the vector characterizing the likelihoods, such that its i-th component is $Pr(y_t | x_t = i)$. Using simple algebra and Bayes' theorem, it is possible to show that the posterior π_t is given as,

$$\pi_t = \tau \text{diag}(L_y) P(t)^T \pi_{t-1}, \tag{3.19}$$

where τ is a normalizing constant that ensures that π_t is a valid mass function, its components summing to unity.

While estimating the posterior mass function is straightforward in the case of a finite state space, in many cases, it is time consuming to evaluate the posterior mass completely. In such cases, we are interested in inferring only the state sequence with the highest likelihood or the maximum a posteriori state sequence. The *Viterbi* algorithm [71, 163] allows the efficient computation of the MAP state sequence. However, in this monograph, we restrict ourself to Bayesian inference of the entire posterior density. We direct the interested reader to an excellent review of hidden Markov models by Rabiner and Juang [162]. We show a simple illustration of an HMM in Figure 3.3.

[1] The posterior probability density in this setting becomes a probability mass function (pmf) as the state space is discrete.

3.4.2 Particle Filters: Monte Carlo Methods

Particle filters [56, 80, 130] approach the Bayesian inference problem by foregoing the requirement for an analytic solution, instead approximating the posterior π_t with a discrete set of particles or samples $\{x_t^{(i)}\}_{i=1}^N$ with associated weights $\{w_t^{(i)}\}_{i=1}^N$ suitably normalized so that $\sum_{i=1}^N w_t^{(i)} = 1$. The approximation for the posterior density is given by:

$$\hat{\pi}_t(x_t) = \sum_{i=1}^{N} w_t^{(i)} \delta_{x_t}\left(x_t^{(i)}\right), \tag{3.20}$$

where $\delta_{x_t}(\cdot)$ is the Dirac delta function centered at x_t. The set $S_t = \{x_t^{(i)}, w_t^{(i)}\}_{i=1}^N$ is the weighted particle set that represents the posterior density at time t, and is estimated recursively from S_{t-1}. The initial particle set S_0 is obtained from sampling the prior density $\pi_0 = p(x_0)$.

We first discuss the *importance function* $g(x_t|x_{t-1}, y_t)$, an easy to sample function whose support encompasses that of π_t. The estimation of $I(f_t)$, as defined in Equation (3.14) can be recast as follows,

$$\begin{aligned} I(f_t) &= \int f_t(x_t) \tfrac{p(x_t|y_{1:t})}{g(x_t|x_{t-1},y_t)} g(x_t|x_{t-1}, y_t) dx_t, \\ &= \int f_t(x_t) w(x_t) g(x_t|x_{t-1}, y_t) dx_t, \end{aligned} \tag{3.21}$$

where $w(x_t)$ is called the *importance weight*,

$$w_t = \frac{p(x_t|y_{1:t})}{g(x_t|x_{t-1}, y_t)}. \tag{3.22}$$

Particle filters sequentially generate S_t from S_{t-1} using the following steps,

1. **Importance sampling:** Sample $x_t^{(i)} \sim g(x_t|x_{t-1}^{(i)}, y_t), i = 1, \ldots, N$. This step is also termed the *proposal step* and $g(\cdot)$ is sometimes called the *proposal density*.
2. **Computing importance weights:** Compute the unnormalized importance weights $\tilde{w}_t^{(i)}$,

$$\tilde{w}_t^{(i)} = w_{t-1}^{(i)} \frac{p(y_t|x_t^{(i)}) p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{g(x_t^{(i)}|x_{t-1}^{(i)} y_t)}, \; i = 1, \ldots, N. \tag{3.23}$$

3. **Normalize weights:** Obtain the normalized weights $w_t^{(i)}$,

$$w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{j=1}^{N} \tilde{w}_t^{(j)}},\ i = 1, \ldots, N. \tag{3.24}$$

4. **Inference estimation:** An estimate of the inference $I(f_t)$ is given by:

$$\hat{I}_N(f_t) = \sum_{i=1}^{N} f_t(x_t^{(i)}) w_t^{(i)}. \tag{3.25}$$

This is performed sequentially to obtain the posterior at each time step. This algorithm suffers from the problem that, after a few time steps, all importance weights, except a few, reduce to zero. These weights will remain at zero for all future time instants (as a result of Equation (3.23)), and do not contribute to the estimation of $\hat{I}_N(f_t)$. Practically, this degeneracy is undesirable and wastes computational resources, but can be avoided with the introduction of a *resampling* step. Resampling essentially replicates particles with higher weights and eliminates those with lower weights. This can be accomplished in many ways, as discussed in [56, 80, 131]. The most popular solution, originally proposed in [80], samples N particles from the set $\{x_t^{(i)}\}$ (samples generated after proposal) according to the multinomial distribution with parameters $w_t^{(i)}$ to produce a new set of N particles $\tilde{S}_t$. The next iteration uses this new set $\tilde{S}_t$ for sequential estimation.

3.4.2.1 Choice of Importance Function

Crucial to the performance of the filter is the choice of the importance function $g(x_t|x_{t-1}y_t)$. Ideally, the importance function should be close to the posterior. If we choose $g(x_t|x_{t-1}y_t) \propto p(y_t|x_t)p(x_t|x_{t-1})$, then the importance weights w_t are identically equal to 1, and the variance of the weights would be zero. For most applications, this density function is not easy to sample from. This is largely due to the non-linearities in the state transition and observation models. One popular choice uses the state transition density $p(x_t|x_{t-1})$ as the importance function, where

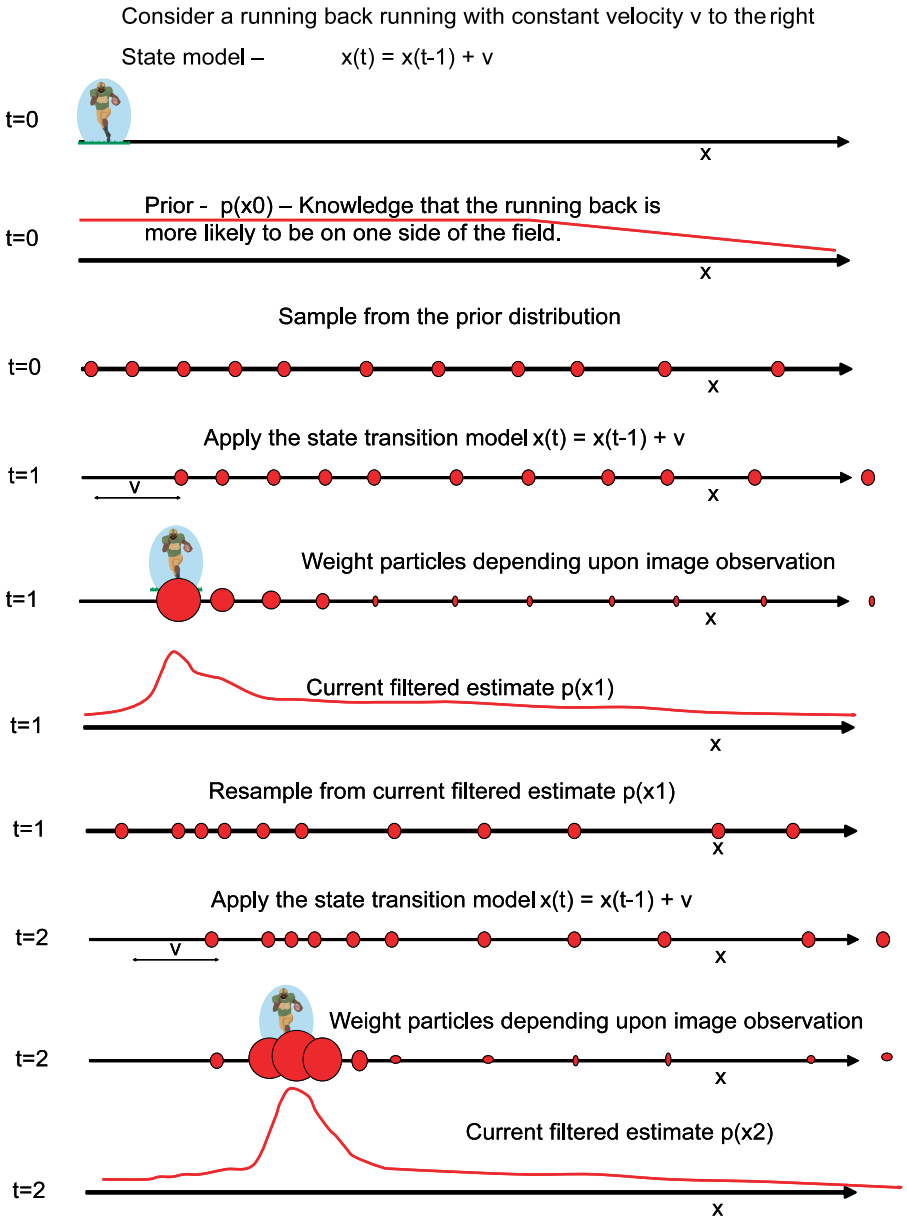


Fig. 3.4 An example illustrating how particle filtering may be used to track moving objects in a video sequence. In this example, the location of the object is the 'state' while the observed video sequence acts as 'observations'. A set of weighted particles are used to maintain the distribution of object location in each frame.

the importance weights are given by:

$$w_t \propto w_{t-1} p(y_t|x_t). \tag{3.26}$$

Other choices include using cleverly constructed approximations to the posterior density [54].

3.4.2.2 Resampling Algorithms

In the particle filtering algorithm, the resampling step was introduced to address degeneracies due to the skewing of the importance weights. Ideally, the variance of the weights should be as low as possible, so that the proposal density closely approximates the posterior density. However, *the unconditional variance of the importance weights can increase with time* [54]. This means that after a few time steps, all but a few of the *normalized weights* are close to zero, and only a fractional part of the sample set S_t contributes to the inference. Among resampling algorithms, the systematic resampling (SR) technique is often used. The basic steps of SR [131] are recounted below.

- For $j = 1, \ldots, N$
  1. Sample $J \sim \{1, \ldots, N\}$, such that $Pr[J = i] = a^{(i)}$, for some choice of $\{a^{(i)}\}$.
  2. The new particle $\tilde{x}_t^{(j)} = x_t^{(J)}$ and the associated weight is $\tilde{w}_t^{(j)} = w_t^{(J)} / a_t^{(J)}$.
- The resampled particle set is $\tilde{S}_t = \{\tilde{x}_t^{(i)}, \tilde{w}_t^{(i)}\}_{i=1}^N$.

If $a^{(i)} = w_t^{(i)}$ the resampling scheme is the one used in [80]. Other choices are discussed in [131]. Particle filtering algorithms that use sequential importance sampling (SIS) and SR are collectively called sequential importance sampling with resampling (SISR) algorithms. Figure 3.4 shows the use of particle filtering to track the location of an athlete in a video. The athlete's location is the 'state' while the observed video sequence acts as the 'observation'. A set of weighted particles maintain the distribution of object location in each frame.

4

Detection, Tracking, and Recognition in Video

In this section, we discuss the main statistical models and algorithms used in key surveillance tasks of detection, tracking, and recognition. A detailed survey of such algorithms for distributed camera networks can be found in [179]. Specifically, we shall examine how the statistical models and inference methods described in the previous chapter play an important role in these applications. Specifically, we discuss how the problem of tracking can be posed as a problem of dynamic state inference, where the state can correspond either to location, identity, behavior, or some combination of these. We discuss how the tools for dynamical inference such as HMMs and particle filters studied in Section 3 come into play in these varied applications. We also discuss how the knowledge of camera imaging studied in Section 2 leads to principled methods for multi-view fusion of target locations.

4.1 Detection

The foremost task in distributed visual sensing is to detect objects of interest as they appear in the individual camera views [61, 102, 136]. In general, this presents a challenging task because objects belonging to the same class (e.g., humans) can differ significantly in appearance due

to factors such as clothing, illumination, pose, and camera parameters. Object detection in images and video may be achieved using one of two major approaches — a static feature-based characterization of the objects of interest or object motion as a cue to detect objects. The first approach typically involves maintaining a model for the objects of interest, in the form of a set of static features and possibly a model for the spatial relationship between them. Given a test image, object detection proceeds in two steps — finding features in the test image and then whether the set of visible features in the test image indicate the presence of the object in the image. One issue with this approach for video arises from the computational expense where it would be inefficient to perform object detection on each frame of the video given a large number of objects. On the other hand, motion as a cue to locate objects of interest significantly reduces the computational load involved in searching for objects in the scene.

In typical visual sensing scenarios, the objects of interest are those that move. Detection of moving objects is a much easier task, because object motion leads to changes in the observed intensity at the corresponding pixel locations. The challenge in a single camera setup is to associate groups of coherently moving nearby pixels to a single object. In multi-camera networks, it also becomes necessary to associate detected objects across camera views.

4.1.1 Background Subtraction

Detection of moving objects is typically performed by modeling the static background and looking for regions in the image that violate this model. The simplest model is that of a single template image representing the static background. A test image can then be subtracted from the template, and pixels with large absolute differences can be marked as *moving*. This simple model introduces the idea of background subtraction — the process of removing static background pixels from an image — so that the pixels associated with moving objects may be highlighted.

Traditionally, background subtraction is posed as a hypothesis test [35] at each pixel, where the *null* hypothesis H_0 is that the pixel

belongs to the background model B_t, while the *alternate* hypothesis H_1 is that the pixel *does not* belong to B_t. Here, the subscript t is used to denote time, B_t represents the background model at time t, and I_t the image at time t. Note that the background B_t may be time-varying, either due to slowly varying environmental factors such as illumination, time of day, shadows etc., or due to rigid camera motion. The hypothesis test

$$\begin{aligned} H_0 &: I_t^i \in B_t^i \text{ (pixel } i \text{ is background)} \\ H_1 &: I_t^i \notin B_t^i \text{ (pixel } i \text{ is NOT background)} \end{aligned} \tag{4.1}$$

has likelihood ratio

$$\frac{\Pr(I_t^i|B_t^i)}{1-\Pr(I_t^i|B_t^i)} \gtrless_{H_1}^{H_0} \tau. \tag{4.2}$$

Here, I_t^i and B_t^i correspond to the i-th pixel of the image and background model, respectively, and τ defines the threshold whose value is chosen according to a desired false-alarm or mis-detection rate. The likelihood ratio test of Equation (4.2) is equivalent to:

$$\Pr(I_t^i|B_t^i) \gtrless_{H_1}^{H_0} \eta = \frac{\tau}{1+\tau}. \tag{4.3}$$

As an example, consider a simple static background, where $B_t = B_0$ is an object-free static background image. For common statistical models, the likelihood ratio test takes the form:

$$|I_t^i - B_0^i| \lessgtr_{H_1}^{H_0} \eta'. \tag{4.4}$$

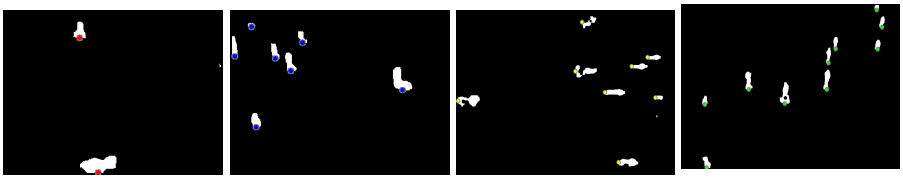
The intuition behind this test is that the error term $|I_t^i - B_0^i|$ is small at pixels that correspond to static objects, while it is large for pixels corresponding to moving objects. Figure 4.1 shows an observed image, the corresponding background model, and the result of background subtraction. As seen, this difference is minimal except in locations corresponding to the moving person and the moving car.

4.1.1.1 Common Background Models

A simple background model such as a fixed template ($B_t = B_0$) would be susceptible to global changes in the environment due to lighting,

(a) Video frames captured from four different views

(b) Background images corresponding to each view



(c) Background subtraction results at each view

(d) Projection of detected points onto synthetic top view of ground-plane.

Fig. 4.1 Use of geometry in multi-view detection. (a) Snapshot from each view, (b) object-free background image, (c) background subtraction results, and (d) synthetically generated top view of the ground plane. The bottom point (feet) of each blob is mapped to the ground plane using the image plane to ground plane homography. Each color represents a blob detected in a different camera view. Points of different colors close together on the ground plane probably correspond to the same subject seen via different camera views. These results were first reported in [179].

time of the day, weather effects, etc. Ideally, we would like the detection algorithm to be:

- Adaptive to global changes in the scene, such as illumination or camera jitter.

- Resilient to periodic disturbances (such as tree movement due to wind or rain).

A host of algorithms are available that work well in many scenarios [61, 102, 136, 196, 224]. One simple extension of the fixed template model is to model each pixel as Gaussian distributed. Given a set of training intensity values for a pixel, $x_1, x_2, \cdots, x_N$, we can use the sample statistics to estimate both the mean and variance at the pixel as (see Section 3),

$$b = \frac{1}{N}\sum_{j=1}^{N} x_j, \quad \sigma^2 = \frac{1}{N-1}\sum_{j=1}^{N}(x_j - b)^2. \tag{4.5}$$

The background model is now defined as:

$$\Pr(I_t^i | b_i, \sigma_i^2) \propto \exp\left(-\frac{(I_t^i - b_i)^2}{2\sigma_i^2}\right). \tag{4.6}$$

The model defined in Equation (4.6) extends the simple static background model by allowing for a variance term σ_i^2 at each pixel. Effectively, this corresponds to using a different threshold at each pixel, one that depends on the variance σ_i^2. Such models are useful in handling dynamic scenes with clutter.

In addition to parametric models, one can employ non-parametric models for learning background models as described in [61]. Given a set of frames $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$, the background model is defined as:

$$p(x) = \frac{1}{N}\sum_{j=1}^{N} K_\sigma(x - x_j), \tag{4.7}$$

where $K_\sigma(\cdot)$ is a kernel function (typically, Gaussian with variance parameter σ). Such non-parametric models are extremely useful to capture complex patterns at each pixel, such as movement of trees or water ripples. As before, background subtraction is performed at each pixel by thresholding the likelihood of the observed intensity.

4.1.1.2 Adaptative Background Model

There are many cases in practice when it becomes necessary to keep the background model tuned to changes in background. Examples include

a global illumination change, and when a previously moving object becomes stationary for a long period of time. An adaptive model for the background is required in these cases, for which various strategies have been proposed. The main focus of these strategies is to update the model in response to changes in background while keeping the background model *target-free*. This is usually possible because, while changes in background have low temporal and spatial frequencies, the variations induced by foreground targets have higher spatial and temporal frequency. A two-step strategy for adapting the background model is proposed in [102]. A moving average $m_{i,t}$ of the observed intensities is computed at each pixel as:

$$m_{i,t} = \alpha m_{i,t-1} + (1-\alpha) I_t^i, 0 < \alpha < 1. \tag{4.8}$$

The moving average $m_{i,t}$ keeps tracks of the truly static components of the scene, as well as slowly changing scene parts (for example, a vehicle that is stationary for a long time or a slow illumination change). This can be used as a template to identify pixels which are possible foreground regions using a simple thresholding of $|I_t^i - m_{i,t}|$. If a pixel is declared non-foreground, then we can update its mean and variance as:

$$b_{i,t} = \beta b_{i,t-1} + (1-\beta) I_t^i, \sigma_{i,t}^2 = \beta(\sigma_{i,t}^2 + b_{i,t-1}^2) + (1-\beta)(I_t^i)^2 - b_{i,t}^2. \tag{4.9}$$

The mean $b_{i,t}$ and variance $\sigma_{i,t}^2$ are now used as the background model for the next frame I_{t+1}^i.

4.2 Tracking

After detecting the objects of interest, the next task is to track each detected object. Most algorithms maintain an appearance model for the detected objects, and use it in conjunction with a motion model for the object, to estimate the object position in each individual camera. Such tracking can be achieved using deterministic approaches that pose tracking as an optimization problem [45, 86] or using stochastic approaches that estimate the posterior distribution of the object location using Kalman filters [31] or particle filters [55, 98, 116, 130, 235]. For surveys on visual tracking of objects, we refer the interested reader to [94, 226].

In many applications, presence of multi-camera inputs allows for the robust estimation of human body pose and limb geometry (markerless motion capture) [74, 50, 199]. When targets have a lower resolution, position tracking in scene coordinates provides useful information for higher level reasoning of the scene. As an example, Xu et al. [225] use multiple cameras to track football players on a ground plane. Similarly, [24, 115, 118, 143, 177] consider the problem of multi-view tracking in the context of a ground plane, with the intent of localization of target position on the plane.

We next discuss an appearance-based tracking algorithm [235], in terms of the statistical models used to describe the underlying dynamical system. Recall that a dynamical system is characterized in terms of three main components: (a) the prior density used to initialize the system, (b) the state transition model, and (c) the observation model. For tracking algorithms, the prior density is an embedding of the detection results onto the state space of the model. The state transition model describes the state evolution, and the observation model is typically an appearance model (in feature space) of the target.

4.2.1 Motion Modeling

To start with, the state space is modeled as the space of affine deformations of a basic shape (in this case, a rectangle). The space of affine deformations forms a 6D Euclidean space. Let $\mathbf{X}_t \in \mathbb{R}^6$ be the 6D affine deformation parameters at time t. They use an adaptive motion model for the state transition,

$$\mathbf{X}_t = \mathbf{X}_{t-1} + \mathbf{v}_t + \mathbf{n}_t, \tag{4.10}$$

where $\mathbf{n}_t$ is a zero-mean Gaussian noise and $\mathbf{v}_t$ is a velocity component computed using first-order frame differencing [235].

4.2.2 Appearance Models

The observation model is a multiple template model consisting of three adaptive templates termed *Stable* ($\mathcal{S}$), *wandering* ($\mathcal{W}$), and *fixed* ($\mathcal{F}$). Each template consists of a mean and variance information

modeled independently for each pixel of the template. In combination, the appearance model at time t is the collection $A_t = \{m_T(t), \mu_T(t), \sigma_T^2(t)\}, T = (\mathcal{S}, \mathcal{W} \text{ and } \mathcal{F})$, where $m(t)$ is the mixture strength at time t. The overall appearance model has a mixture of Gaussian distribution, parametrized by the mixture means $\mu(t)$ and variances $\sigma^2(t)$, and mixing weights of the three templates. Hence, the observation model is given as,

$$p(\mathbf{y}_t|\mathbf{X}_t) = p(\mathbf{y}_t|\mathbf{X}_t, A_t) = p(\mathbf{z}_t = \mathcal{T}(\mathbf{y}_t, \mathbf{X}_t)|A_t), \tag{4.11}$$

where $\mathbf{z}_t = \mathcal{T}(\mathbf{y}_t, \mathbf{X}_t)$ is the region on the image $\mathbf{y}_t$ corresponding to the state $\mathbf{X}_t$. The density $p(\mathbf{z}_t|A_t)$ is the pixel-wise mixture of Gaussian distribution, with the means, variances, and mixture strengths as defined in the appearance model A_t.

The three mixture components — *stable, wander*, and *fixed* — are adaptively updated using different strategies. The *stable* model is updated using a moving average of the tracked region. This allows it to be to robust to static features of the template. The *wander* component corresponds to the MAP estimate of the tracking appearance at the previous time instant. This component tracks transient changes in the appearance, such as a quick pose change of the head or a quick illumination change. The *fixed* component is static and not updated, and typically corresponds to a nominal appearance such as the frontal face. Details of this update strategy can be found in [101, 235]. In Figure 4.2, we show some results of tracking of an individual using the algorithm described above.

4.3 Multi-View Metric Estimation

In Section 2, we described geometric constraints that arise in multi-view problems. In particular, we are interested in the homography transformation that is induced in the presence of a plane. The homography transformation provides an example of an invertible projective transformation. Geometric properties such as invariants have been well studied [89, 137]. However, the geometric theory analyzing the properties of the projective transformation usually operates under the assumption of noise-free measurements, or employs heuristics to

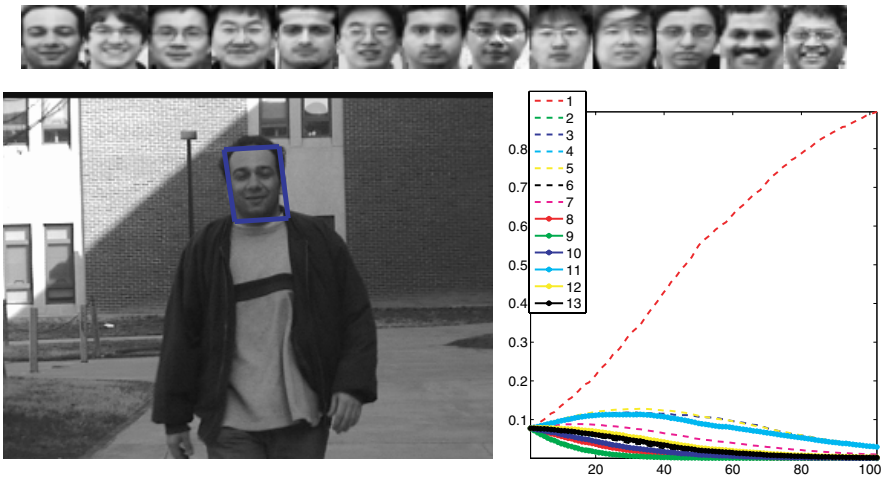


Fig. 4.2 Simultaneous tracking and recognition of faces from a video sequence. (top) Face templates of the 13 persons making the gallery set, (left) snapshot of tracking results, and (right) probability scores for recognition, $p(\theta_t|y_{1:t})$. Here the index for identity corresponds to the same as the ordering of faces in the gallery. These results were first reported in [234].

account for noise. A rigorous and formal characterization of statistical estimation and projective transformations has not been adequately developed.

Kanatani [109] pioneered the research of statistical estimation using the structure induced by geometry for various estimation problems in 2D and 3D. Kanatani [109] discusses the role of geometric constraints in solving line, plane, and conic fitting problems using covariance matrices for error characterization. In [46], Criminisi uses similar error characterization (in terms of covariance matrices) for metrology in multi-view settings. In [72], the choice of representation for projective entities is considered in the light of earlier work by Kanatani and others. Linearization approaches for propagating covariance matrices through projective transforms are used in [148] for localization and mosaicking.

Hartley and Sturm [90] addressed the problem of two-view triangulation for the case of image-plane measurements corrupted with Gaussian noise. Traditionally, the 3D location of a point is obtained by minimizing a cost function over the world coordinates. It is argued

that such a cost function might not be appropriate if the scene reconstruction deviates from Euclidean by a projective transformation and suggests minimization over image plane coordinates where the probability distributions are more meaningful. In the next section, we discuss the effect of camera projection on random variables in the real world. Then, we discuss how this is useful for two applications — multi-view tracking and estimating heights of objects from multiple views.

4.3.1 Random Variables under Projective Transformations

Applications dealing with the estimation of real-world metrics like lengths and location from multi-view inputs (see Figure 4.3) involve a projective transformation of noisy image plane measurements. Before we devise a strategy for fusion, we need to study the properties of the estimates produced by each camera. Clearly, the error in tracking of the object on its image plane influences the statistical properties of the estimate from a given camera. However, the homography transformation between the image plane of the camera and the ground plane plays a larger role (see Section 2.2 for details on projective imaging).

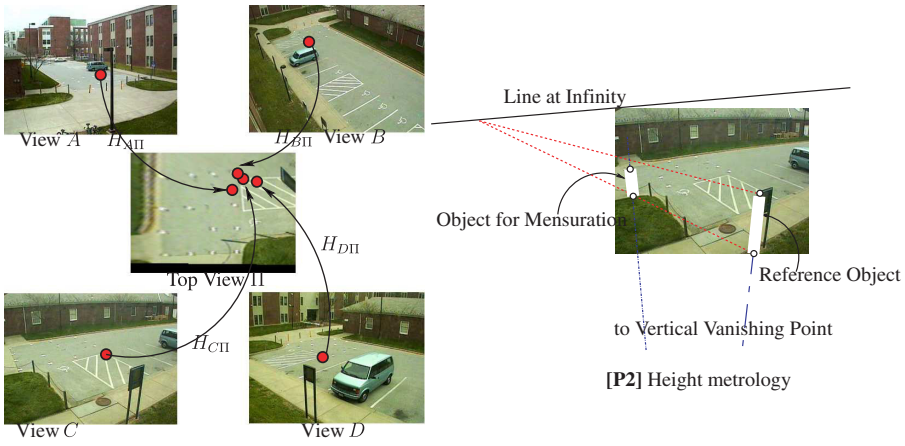


Fig. 4.3 Depiction of problems that critically involve projective transformations. (left) [P1] Multi-view fusion of object locations on a plane can possibly involve image plane location estimates projected through different transformations. Fusion of these ground plane estimates requires knowledge of the statistics of the transformed estimates. (right) [P2] Metrology of object heights using cross-ratios involves a 1D projective transformation.

The homography transformations are bound to differ given changing views, given that each camera is guaranteed to have its own external parameters with respect to the world coordinate system. Hence, a different homography generates each image plane estimate. Therefore, *even if the properties of the noise on the image plane are identical across camera views, the statistical properties of the transformed estimate are in general different.* A robust fusion scheme must necessarily account for the different statistical properties of the individual estimates.

In [177], it is shown that the distribution of a projectively transformed random variable does not have well-defined moments. However, when the imaged region of interest is distant from the line at infinity (of the transformation), we can assume the transformed random variable to have well-defined moments. This provides a theoretical basis for understanding the poor performance of ground plane tracking algorithms near the horizon line. The theory finds application in metric estimation from multi-view observations. In particular, we highlight its use in two applications: multi-view tracking and multi-view metrology for height estimation.

4.3.2 Multi-view Tracking

The application of multi-view tracking benefits immensely from a statistical characterization of projectively transformed random variables. Many proposed tracking algorithms use a planar scene assumption to track objects in the real world. The single-camera multiple-human tracking algorithm in [232] uses the ground plane motion constraint to estimate foot location on the ground plane to decide depth ordering. The algorithm in [231] uses a Kalman filter to track the location and velocity on the plane with the observation noise model obtained by linearization of the homography between the image and the ground plane. An alternate single camera tracker using the homography to project head and feet positions to the ground plane is described in [67]. Multi-camera tracking algorithms [69, 115, 118] also use homographies to project inputs from background subtraction onto the ground plane.

Typically, data association and target localization are achieved through consensus among projections from the cameras. As an example,

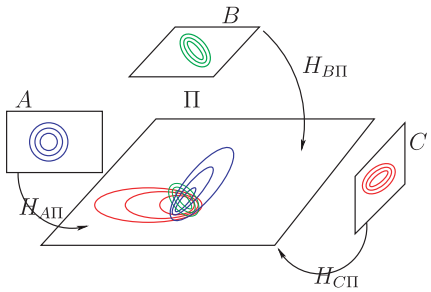


Fig. 4.4 A schematic showing densities on the image planes of cameras and their transformations to the ground plane. The variance in position estimate at each of the individual cameras (A, B, and C) is represented using colored circles. These variances when projected onto the ground plane lead to highly anisotropic distributions, thereby requiring an estimator that takes account of this transformation. If this transformation is ignored, then the fused estimator will be biased. See Figure 4.6 for an example of this bias leading to track association errors. Image courtesy [177].

consider a point object being imaged onto multiple views. Due to imaging and modeling/estimation inaccuracies, the locations of points on the individual image planes are not accurately known. Even assuming identical individual image plane error variances, the variance on the ground plane from the individual views can differ greatly. Finally, such variations depend on the actual location of the imaged point. Figure 4.4 explains this concept graphically. Suppose, we are interested in an algorithm to fuse the individual estimates. It is immediately clear that estimates such as the sample mean will not be *efficient*, as the individual estimates are not identically distributed. In [177], the accuracy of the transformed estimate is shown to be significantly dependent on the line at infinity of the projective transformation between a camera view and the ground plane. Image plane estimates that lie near the line at infinity might have ill-defined moments (either in terms of high variance or in the existence of the moments itself) for the transformed ground plane estimates.

4.3.2.1 Static Localization

Given M cameras, and the homography matrices $H_i, i = 1, \ldots, M$ between the camera views and the ground plane, we describe an algorithm for fusing location estimates. Let $\mathbf{Z}_U^i$ be the random variable

modeling the target location on the image plane of the i-th camera. We first assume that the random variables $\{\mathbf{Z}_U^i\}_{i=1}^M$ are statistically independent. This assumption is justified for imaging (sensor) noise. However, for cases such as occlusion and parallax, the 'noise' is due to error in modeling. In such cases, the noises/errors are correlated across cameras, and estimation of this correlation is complicated.

The distribution of $\mathbf{Z}_U^i$ is generated by a tracking algorithm or a detection algorithm. However, we are only interested in the mean $\mathbf{m}_u^i$ and the covariances S_u^i of the distribution. If the underlying tracker is indeed a Kalman or a particle filter, then we can readily obtain estimates of mean and the covariances as output. In cases where this is not possible, such as the Kanade–Lucas–Tomasi (KLT) tracker, we assume the mean to be the tracker output itself and suitable values for the covariance matrices.

We project the random variables from the image plane to the ground plane to obtain the transformed variables $\{\mathbf{Z}_X^i\}$ such that $\mathbf{Z}_X^i = H_i(\mathbf{Z}_U^i)$. We use the unscented transformation [104, 105] to estimate the mean $\hat{\mu}_x^i$ and covariance matrix $\hat{\Sigma}_x^i$ of $\mathbf{Z}_X^i$. We can now formulate a minimum variance estimator for the location on the ground plane. Further, it is a simple exercise [177] to show that the estimates $\hat{\mu}_x^i$ are unbiased ($\mathrm{E}(\hat{\mu}_{\mathbf{x}}^i) = \mu_{\mathbf{x}}$ the true target location). The minimum variance estimate $\hat{\mu}_{\mathbf{x}} = (\hat{\mu}_x, \hat{\mu}_y)^T$ is computed as [186, 187]:

$$\hat{\mu}_{\mathbf{x}} = \sum_{i=1}^{M} (\hat{\Sigma}_x^i)^{-1} \Sigma_{mv} \hat{\mu}_x^i, \Sigma_{mv} = \left(\sum_{j=1}^{M} (\hat{\Sigma}_x^j)^{-1} \right)^{-1}. \tag{4.12}$$

Finally, the covariance matrix of the minimum variance estimate $\hat{\mu}_{\mathbf{x}}$ can be computed from Equation (4.12) as

$$\mathrm{covar}(\hat{\mu}_{\mathbf{x}}) = \Sigma_{mv} = \left(\sum_{j=1}^{M} \left(\hat{\Sigma}_x^j \right)^{-1} \right)^{-1}. \tag{4.13}$$

Given the homography matrices for a set of cameras observing a plane, we can compute and plot the variance of the minimum variance estimator as a function of the true mean μ on the ground plane. Figure 4.5 shows one such plot.

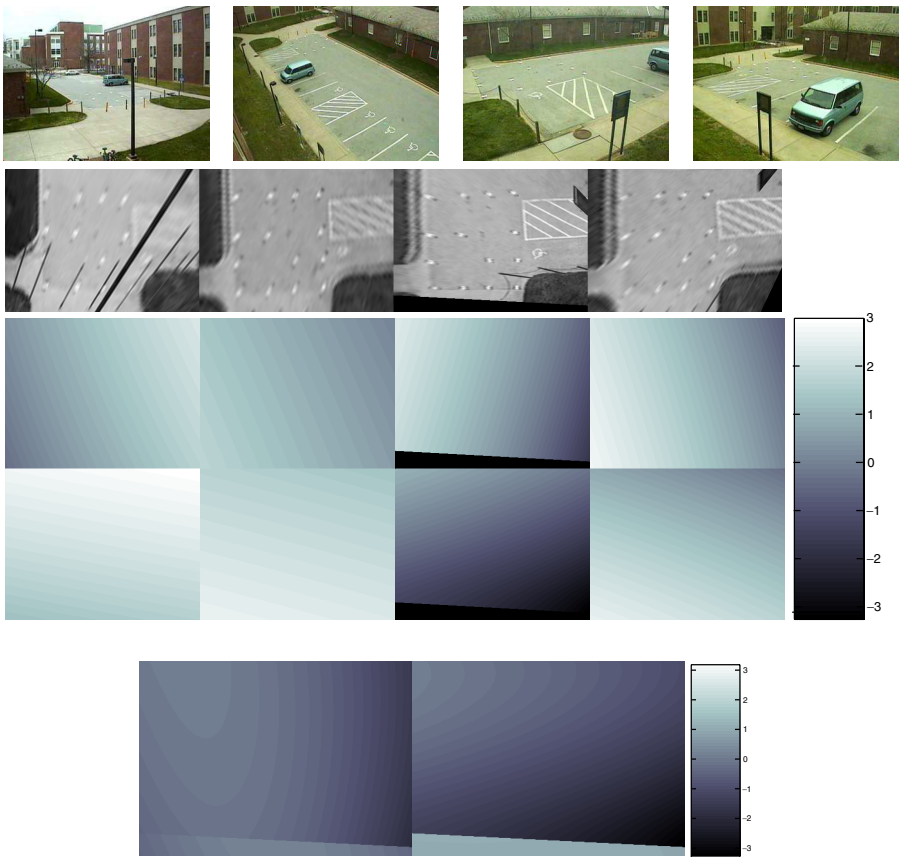


Fig. 4.5 Variance estimates of the camera setup shown in the top row. (top row) Camera views, (second row) top view of the plane generated from the corresponding view from the first row, and (third row) variance estimate of Z_x and Z_y for each camera in log scale over the ground plane. (last row) Variance (in log scale) of the minimum variance location estimator along the two axes. These results were first reported in [177].

4.3.2.2 Dynamical System for Tracking

In most cases, we are interested in tracking the location with time (and not just a static estimation), such as in images captured by a video camera recording a pedestrian. We formulate a discrete time dynamical system for location tracking on the plane. The state space comprises the location and velocity on the ground plane. Let $\mathbf{x}_t$ be the state space at time t, $\mathbf{x}_t = [x_t, y_t, \dot{x}_t, \dot{y}_t]^T \in \mathbb{R}^4$. The state evolution is defined using

a constant velocity model,

$$\mathbf{x}_t = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{t-1} + \omega_t, \tag{4.14}$$

where ω_t is a noise process. The observation vector $\mathbf{y}_t \in \mathbb{R}^{2M}$ is the stack of location means estimated from each camera using the unscented transformation. The observation model is given as,

$$\mathbf{y}_t = \begin{bmatrix} \hat{\mu}_{\mathbf{1}} \\ \vdots \\ \hat{\mu}_{\mathbf{M}} \end{bmatrix}_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \vdots & & & \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{x}_t + \Lambda(\mathbf{x_t})\Omega_t, \tag{4.15}$$

where Ω_t is a zero-mean noise process with an identity covariance matrix. $\Lambda(\mathbf{x_t})$ determines the covariance matrix of the overall noise as,

$$\Lambda(\mathbf{x_t}) = \begin{bmatrix} \Sigma_x^1(\mathbf{x_t}) & \cdots & 0_{2\text{X}2} \\ \vdots & \ddots & \vdots \\ 0_{2\text{X}2} & \cdots & \hat{\Sigma}_x^M(\mathbf{x_t}) \end{bmatrix}^{\frac{1}{2}}, \tag{4.16}$$

where $0_{2\times 2}$ is a 2×2 matrix with zero entries, and $\Sigma_x^i(\mathbf{x_t})$ is the covariance matrix of $\mathbf{Z}_X^i$ when the true location of the target on the ground plane is $\mathbf{x_t}$. The model of Equation (4.15) has two important properties.

- The noise properties of the observations from various views differ, and the covariances depend not only on the view, but also on the true location of the target $\mathbf{x}_t$. This dependence is encoded in Λ.
- The MLE of $\mathbf{x}_t$ (i.e., the value of $\mathbf{x}_t$ that maximizes the probability $p(\mathbf{y}_t|\mathbf{x}_t)$) is the minimum variance estimator described in the previous sub-section.

Tracking of target(s) can now be performed using a Kalman or a particle filter. A multi-target tracking system was designed to test the efficacy of the proposed models with the camera placement of Figure 4.5.

The bottom-most point from each background subtracted blob at each camera is extracted and projected onto the world plane. Association of these data to trackers is performed using the classical joint probability data association filter (JPDAF) [9], with data from each camera associated separately. Ground truth was obtained using markers. As before, two observation models are compared: one employs the proposed approach and the other assumes isotropic modeling across views. Finally, a particle filter is used to track targets. Testing was performed with a video of 8,000 frames that had three targets introduced sequentially at frames 1,000, 4,300, and 7,200. Figure 4.6 shows tracking results for this experiment. The proposed model consistently results in lower KL divergence to the ground truth.

4.3.3 Metrology

The most common way to measure lengths from images of objects and scenes involves the use of cross-ratios and the vanishing lines of a plane along with the vertical vanishing point [47]. Figure 4.7 illustrates this.

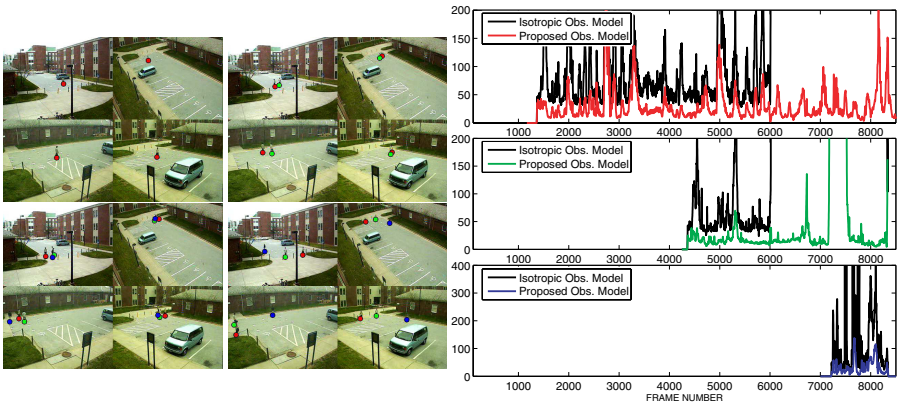


Fig. 4.6 Tracking results for three targets with the four camera data set. (best viewed in color/at high zoom) (left) Snapshots of tracking output at various timestamps. (right) Evaluation of tracking using symmetric KL divergence from ground truth. Two systems are compared: one using the proposed observation model and the other using isotropic models across cameras. Each plot corresponds to a different target. The trackers using isotropic models swap identities around frames 6,000. The corresponding KL-divergence values go off scale. These results were first reported in [177].

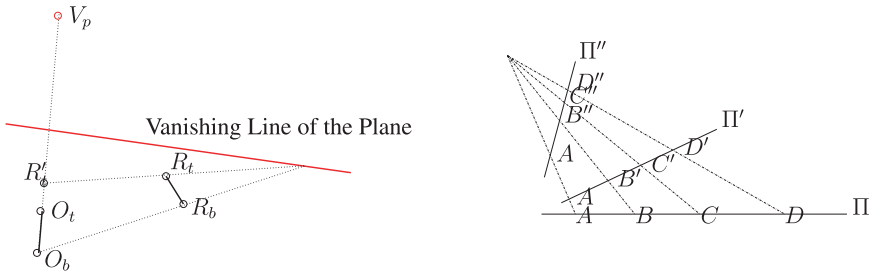


Fig. 4.7 (Left) **Metrology:** Using a reference object R_bR_t with known length h_r to measure the length of O_bO_t. Both objects are assumed to be oriented vertical to the plane. Knowledge of the vanishing line and the vertical vanishing point V_p is required. (right) **Invariance of the cross-ratio:** The three lines Π, Π', and Π'' are under central projection, hence related by a 1D homography. Given any three point correspondence, this homography can be uniquely determined. The fourth point can no longer be independently chosen on the lines.

Using the invariance of the cross-ratio between the points V_p, O_b, O_t and R'_t, we obtain:

$$\frac{h_o}{h_r} = \frac{|O_tO_b|}{|V_pO_t|}\frac{|V_pR'_t|}{|R'_tO_b|}, \tag{4.17}$$

where h_r is the true height of the reference object.

Lv et al. [135] model the human as a stick of fixed height, perpendicular to the ground plane. The motion of such an object creates virtual parallel lines along the plane normal (feet-to-head) and along the plane (head-to-head and feet-to-feet). This can be used to recover the vanishing line of the plane, as well as the vertical vanishing point.

Shao et al. [182] built an automatic metrology system that estimates the vanishing line and vertical point (similar to [135]) by grouping trajectories that have similar vanishing points. The calibration information is used to obtain a height estimate at each frame. The fusion of this temporal data is done under the framework of stochastic approximation [167] with a least median square cost function to achieve robustness towards outliers. The stochastic approximation methods allow for a principled fusion of the noisy time series data. Figure 4.8 showcases metrology results obtained for the Honeywell data set. For each person (and in each view), the single frame metrology estimates are used

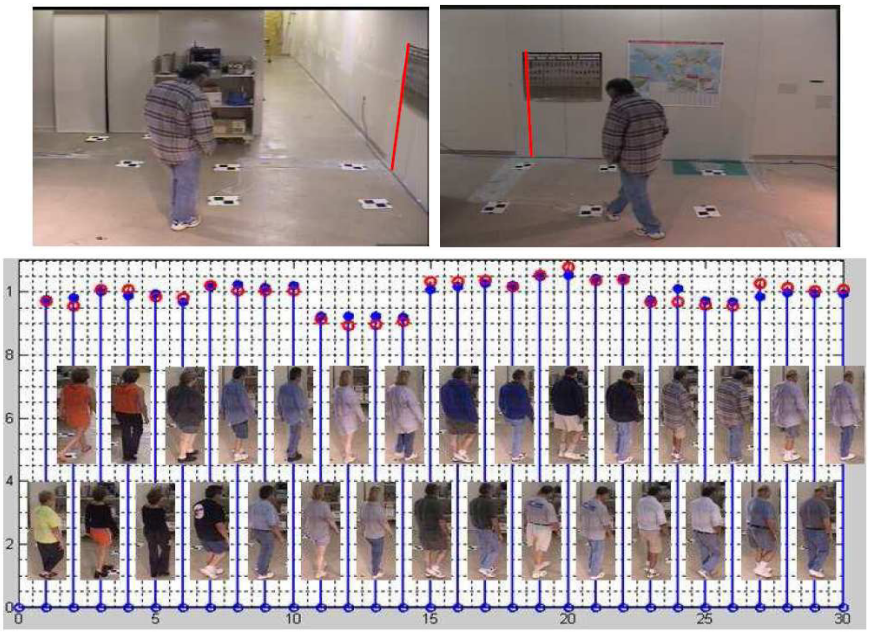


Fig. 4.8 Metrology results on the Honeywell data set. (top row) Example frames from two different views, with the reference objects marked in red. (bottom row) metrology results of 30 pedestrians with each stem representing the final estimate over one set of data. The filled blue and unfilled red circles represent height estimates from the two views. These results were first reported in [182].

using the stochastic approximation methods. The figure shows final height estimates from two different views.

4.4 Behavioral Motion Models for Tracking

Behavioral research in the study of the organizational structure and communication forms in social insects, such as ants and bees, has received much attention in recent years [191, 219]. Such study has provided practical models for tasks such as work organization, reliable distributed communication, navigation, etc [145, 146]. Usually, in experiments to study these insects, the insects in an observation hive are videotaped. Then, hours of video data are manually studied and hand-labeled. This task comprises the bulk of the time and effort in such experiments. In this section, we discuss general methodologies for

automatic labeling of such videos and provide an example by analyzing the movement of bees in a hive. In [215], they rack and recognize the behavior exhibited by the bees simultaneously. In such a joint approach, behavior models act as priors for motion tracking and significantly enhance motion tracking, while accurate and reliable motion tracking enables behavior analysis and recognition.

4.4.1 Anatomical Modeling and State Space for Tracking

Modeling the anatomy of insects is crucial for reliable tracking, because the structure of their body parts and their relative positions present physical limits on their possible relative orientations. In spite of their great diversity, the anatomy of most insects is rather similar. The body is divided into three main parts: the head, thorax, and abdomen. Figure 4.9 shows images of a bee, an ant, and a beetle. Though individual differences occur in their body structure, the three main parts of the body are evident. Each of these three parts can be regarded as rigid body part for the purposes of video-based tracking. Most insects also move in the same direction their head faces. Therefore, during specific movements such as turning, the orientation of the abdomen usually follows the orientation of the head and the thorax with some lag.

The bees are modeled as three connected ellipses, one for each body part. The effect of the wings and legs on the bees is neglected. Figure 4.9 shows the shape model of a bee. The location of the bee

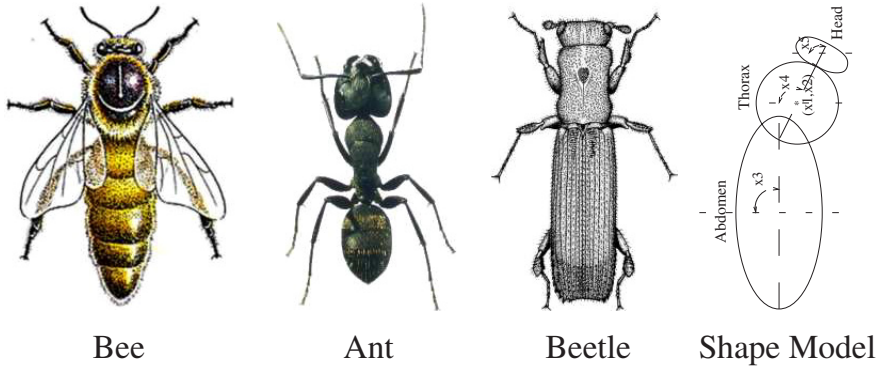


Fig. 4.9 A bee, an ant, a beetle, and the insect shape model. Image courtesy [215].

and its parts in any frame can be given by five parameters: the location of the center of the thorax (two parameters), the orientation of the head, the orientation of the thorax, and the orientation of the abdomen (refer Figure 4.9). Tracking the bee in a video amounts to estimating these five model parameters, $\mathbf{u} = [x_1\ x_2\ x_3\ x_4\ x_5]$ for each frame. This 5-parameter model has a direct physical significance in terms of defining the location and the orientation of the various body parts in each frame.

4.4.2 Behavior Modeling

Insects, especially social insects like bees and ants, exhibit rich behaviors. Modeling such behaviors is helpful in accurate and robust tracking. Moreover, explicitly modeling such behaviors leads to algorithms where position tracking and behavior analysis are tackled in a unified framework. Following the premise that the mixed state space model can be used to generate complicated dynamics, we model the basic motions as a vocabulary of local motions. These basic motions are then regarded as states, and behaviors are modeled as being Markovian on this motion state space. Once each behavior has been modeled as a Markov process, then our tracking system can simultaneously track the position and the behavior of insects in videos.

The probability distribution of location parameters $\mathbf{u}$ for certain basic motions (m_1–m_4) — (1) moving straight ahead, (2) turning, (3) waggle, and (4) motionless — are modeled explicitly. The basic motions, straight, waggle, and motionless are modeled using Gaussian pdfs (p_{m1}, p_{m3}, p_{m4}). A mixture of two Gaussians (p_{m2}) models the turning motion (to accommodate the two possible turning directions).

$$p_{mi}(\mathbf{u}_t|\mathbf{u}_{t-1}) = \mathrm{N}(\mathbf{u}_{t-1} + \vec{\mu_{mi}}, \Sigma_{mi}), \quad \text{for } i \in \{1,3,4\}, \tag{4.18}$$

$$\begin{aligned} p_{m2}(\mathbf{u}_t|\mathbf{u}_{t-1}) &= 0.5\mathrm{N}(\mathbf{u}_{t-1} + \vec{\mu}_{m2}, \Sigma_{m2}) \\ &\quad + 0.5\mathrm{N}(\mathbf{u}_{t-1} - \vec{\mu}_{m2}, \Sigma_{m2}). \end{aligned} \tag{4.19}$$

Each behavior $\theta \in \{1, \ldots, C\}$ is now modeled as a Markov process of order K_θ on these motions, i.e.,

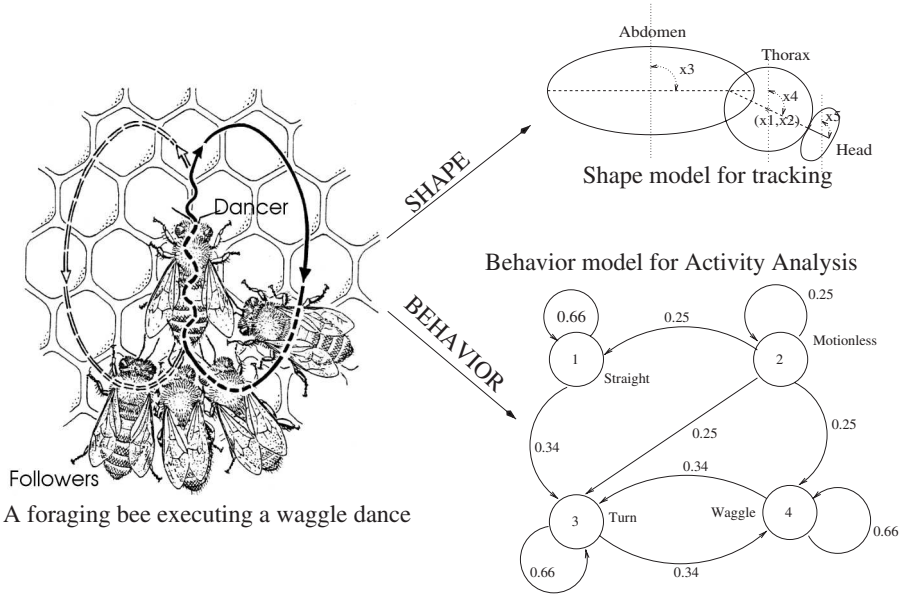
$$\mathbf{s}_t = \sum_{k=1}^{K_\theta} A_\theta^k \mathbf{s}_{t-k}, \tag{4.20}$$

where s_t is a vector whose j-th element is the probability that the bee is in the motion state m_j, and K_θ is the model order for the behavior indexed by θ. The parameters of each behavior model comprised autoregressive parameters A_θ^k for $k = 1, \ldots, K_\theta$.

Three different behaviors — the waggle dance, the round dance, and a stationary bee — are modeled using a first-order Markov model. For illustration, we discuss the manner in which the waggle dance is modeled. Figure 4.10 shows the trajectory followed by a bee during a single run of the waggle dance. It also shows some followers who follow the dancer but do not waggle. A typical Markov model for the waggle dance is also shown in Figure 4.10.

4.4.3 System Modeling

As before, the tracking problem is addressed by estimating the state $\mathbf{X}_t = (\mathbf{x}_t, \theta_t)$ given the image observations $y_{1:t}$. The state $\mathbf{x}_t = (\mathbf{u}_t, \mathbf{s}_t)$ encompasses the basic motion states (encoded in $\mathbf{s}_t$) as well as the



A foraging bee executing a waggle dance

Fig. 4.10 A bee performing a waggle dance and the behavioral model for the waggle dance. Image courtesy [215].

location of the bee on the image plane encoded in $\mathbf{u}_t = (x_1, \ldots, x_5)_t$. The state transition model characterized by the density $p(\mathbf{X}_t|\mathbf{X}_{t-1})$ is defined as,

$$p(\mathbf{X}_t|\mathbf{X}_{t-1}) = p(\mathbf{x}_t|\mathbf{x}_{t-1}\theta_t)p(\theta_t|\mathbf{x}_{t-1}\theta_{t-1}). \tag{4.21}$$

The term $p(\theta_t|\mathbf{x}_{t-1}\theta_{t-1})$ controls the switching of behavior and is learned from training data. The dynamics of tracking encoded in $p(\mathbf{x}_t|\mathbf{x}_{t-1}\theta_t)$ are defined by Equations (4.18)–(4.20). Given the location of the bee in the current frame $\mathbf{u}_t$ and the image observation given by y_t, we first compute the appearance $z_t = \mathscr{T}(y_t, \mathbf{u}_t)$ of the bee in the current frame (i.e., the color image of the three ellipse anatomical model of the bee). The observation model is given as,

$$p(y_t|\mathbf{x}_t\theta_t) = p(y_t|\mathbf{u}_t) = p(z_t|A_1, \ldots, A_5), \tag{4.22}$$

where the $A_1, \ldots, A_5$ are color-templates that form the exemplars for the appearance of the bee. The red, green and blue (RGB) components of color are treated independently and identically. The bee's appearance in any given frame is assumed to be Gaussian centered around one of these five exemplars, i.e.,

$$p(z_t|A_1, \ldots, A_5) = \frac{1}{5}\sum_{i=1}^{i=5} \mathrm{N}(z_t; A_i, \Sigma_i), \tag{4.23}$$

where $N(z; A_i, \Sigma_i)$ stands for the normal distribution with mean A_i and covariance Σ_i.

4.4.4 Experimental Results

Tracking experiments on video sequences of bees in a hive were conducted by [215]. In the videos, the bees exhibited three behaviors: waggle dancing, round dancing, and being stationary. The video sequences ranged between 50 and 700 frames long. It is noteworthy that a similar tracking algorithm without a behavioral model lost track within 30–40 frames. With this behavior-based tracking algorithm, the bees were tracked during the entire length of these videos. Parameters, such as orientations of the various body parts, were also extracted during each

Fig. 4.11 Sample frames from a tracked sequence of a bee in a beehive. Images show the top five particles superimposed on each frame. Blue denotes the best particle and red denotes the fifth best particle. Frame numbers row-wise from top left: 30, 31, 32, 33, 34, and 90. Figure best viewed in color. These results were first reported in [215].

frame of the video sequences. These parameters were then used to identify the behaviors automatically. The estimate was verified manually and was found to be robust and accurate.

Figure 4.11 shows the structural model of the tracked bee superimposed on the original image frame. In this particular video, the bee was exhibiting a waggle dance. As apparent in the sample frames, the appearance of the dancer varies significantly within the video. These images display the tracker's ability to maintain track consistency even under extreme clutter and in the presence of several similar looking bees. Frames 30–34 show the bee executing a waggle dance. Notice that the abdomen of the bee waggles from one side to another.

A portion of the tracking result was validated by comparing it with a ground truth track obtained manually ("point and click") by an experienced human observer. The tracking result obtained using the proposed method approaches close to manual tracking results. The mean differences between manual and automated tracking using our method are given in Table 4.1. The positional differences are small compared to the average length of the bee, which is about 80 pixels (from front of head to tip of abdomen).

Table 4.1. Comparison of our tracking algorithm with ground truth.

	Average positional difference between ground truth and our algorithm
Center of abdomen	4.5 pixels
Abdomen orientation	0.20 radians (11.5 deg)
Center of thorax	3.5 pixels
Thorax orientation	0.15 radians (8.6 deg)

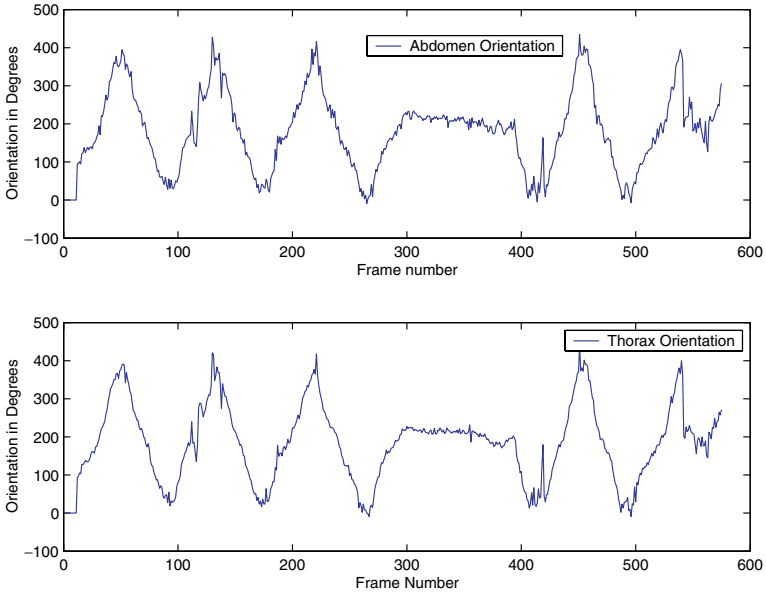


Fig. 4.12 The orientation of the abdomen and the thorax of a bee in a video sequence of approximately 600 frames. These results were first reported in [215].

Figure 4.12 shows the estimated orientation of the abdomen and the thorax in a video sequence of approximately 600 frames. The orientation is measured with respect to the vertically upward direction in each image frame — a clockwise rotation would increase the angle of orientation, and an anticlockwise rotation would decrease the angle of orientation. The waggle dance is characterized by the central waggling portion, which is immediately followed by a turn, a straight run, another turn, and a return to the waggling section as shown in Figure 4.10. The turning direction is reversed after each alternate waggling section. This is clearly seen in the orientation of both abdomen and the thorax. The

sudden change in slope (from positive to negative or vice versa) of the angle of orientation denotes the reversal of turning direction. During the waggle portion of the dance, as the bee moves its abdomen from one side to another, it continues to advance slowly. The large local variation in the orientation of the abdomen just before every reversal of direction shows the waggling nature of the abdomen. Moreover, the average angle of the thorax during the waggle segments denotes the direction of the waggle axis.

4.5 Simultaneous Tracking and Recognition

In this section, we show how the problems of tracking and recognition can be posed as a joint state-estimation problem. This state consists of both the motion and appearance parameters as is usually the case in tracking applications, in addition to the unknown identity parameter. The identity parameter would be used for recognition applications. It is interesting to note that both these disparate applications can be fused together in the Bayesian setting discussed previously. This offers improvements compared to traditional track-then-recognize approaches.

Tracking and Recognition with 2D Appearance Models

A 2D appearance model offers a simple yet effective feature for classification. This consists of a probability distribution in image space for each class. Typically, a parametric family of distributions is chosen, and the parameters of the family are learned for each class separately from the available training data. The most common parametric family is the Gaussian distribution or a mixture of Gaussian distributions.

Such 2D appearance models are a natural choice, specifically for modeling and recognizing planar or near-planar objects (ignoring effects of self-occlusion), because it is fairly simple to account for the effect of viewpoint on these appearance models. In particular, small viewpoint changes produce affine deformations on the 2D appearance models. Thus, affine-invariant 2D appearance models are common and effective representations for recognizing planar and near-planar objects.

Nevertheless, the problem with using a single 2D appearance model occurs when the pose of a 3D object changes. In such a case, a simple 2D appearance model cannot account for this change in pose.

Several approaches use 2D affine-invariant appearance models for simultaneous face tracking and recognition [3, 12, 144, 211, 234, 233]. As an example, let us consider the simultaneous face tracking and recognition framework presented in [234]. First, a simple appearance-based algorithm that uses a particle filter is used to track objects of interest. For each person in the gallery, the set of all images inside the tracked bounding box are used as training samples. These are then used to learn the parameters of the appearance model (mixture weights, mean, and covariance of each Gaussian component). A particle filter is then used to estimate the position of the target's face and the identity of the individual simultaneously.

During testing, when a subject moves in front of the camera, the observed appearance is compared with the models stored in the gallery to recognize the individual. Again, recognition is usually performed by MAP estimation. The top row of Figure 4.2 shows the stored 2D appearance templates for the individuals in the gallery. In the bottom, two images from a test sequence with the bounding box show the location of the target's face. The image within the bounding box is matched with the stored 2D appearance models in the gallery to perform recognition.

3D Face Tracking and Recognition with Geometric Face Models

2D appearance models do not adequately account for the inherent changes in the appearance that occur due to large pose changes in the video (especially for non-planar objects like faces, cars, etc.). For applications such as face tracking and recognition it becomes extremely important to account for pose changes that occur throughout the video to make continuous recognition possible. One way to account for pose changes is to model the face as a 3D object with a certain structure and a corresponding texture. Since the variations in face structure across individuals tend to be modest, one can assume a generic 3D model

for the face, varying the texture according to the individual. The texture forms the identity cue while the 3D generic face model allows recognition to be performed irrespective of the pose. Several competing approaches exist for fitting 3D models to a face for recognition purposes. In [25, 60, 100], a statistical model of 3D faces is learned from a population of 3D scans, and recognition is performed after morphing the acquired image to the 3D model. Unfortunately, moving from a planar model to complicated 3D models also introduces the significant issue of registering the acquired 2D image to the 3D model. As the number of parameters in the 3D model increases, the registration task becomes difficult. Therefore, several approaches have adopted simple parameterized 3D models to model the face, keeping the registration between a 2D image and a 3D model simple.

A simple but effective model for the generic 3D model of a face is that of a cylinder [4, 122]. Such a simple model has the advantage that occlusion analysis becomes facile allowing efficient inference algorithms to estimate the pose at each frame. Once this task is accomplished for each independently, the image intensities in the original image are then mapped onto the generic 3D model to obtain the texture map of the face. The texture-mapped models obtained at each camera node can be fused to obtain a complete 3D model of the face. This texture-mapped model is then compared with the stored texture maps of all the 3D models in the gallery to perform face-based person recognition. In another important point, the face is assumed to be cylindrical, so after estimating the face's pose, the surface normals at each of the points on the face are known. This allows us to extract texture features that are moderately insensitive to illumination conditions. Therefore, modeling the 3D structure of the face to perform simultaneous tracking and recognition allows us to design recognition algorithms that are robust to both changes in facial pose and illumination conditions. Figure 4.13 shows some results [4] of 3D facial pose tracking and recognition. Notice that the pose of the face is accurately estimated in spite of the significant variations in lighting, pose, and occlusions. The graphical rendering in the last column shows the cylindrical face model at the pose estimated from the images in the third column. An implementation of this algorithm suitable for smart cameras is discussed in [174].

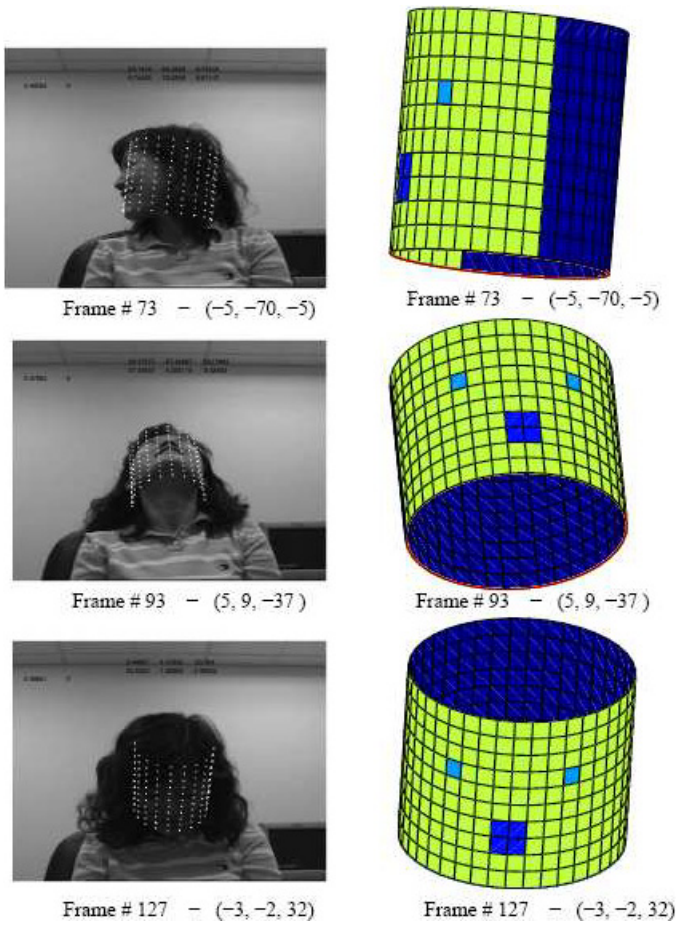


Fig. 4.13 Tracking results under extreme poses and different illumination conditions. The cylindrical grid is overlaid on the image plane to display the results. The three-tuple shows the estimated orientation (roll, yaw, pitch) in degrees. The second column shows a cylindrical model in the pose estimated for the sequence in the first column. These results were first reported in [4].

4.5.1 Local Feature-Based Methods for Object Recognition

In recent years local feature-based methods for object recognition have gained popularity. It is extremely difficult to model the deformation of global features due to changes in viewpoint and illumination conditions, but it is relatively simple to model the local deformations due to these structured changes. Feature-based object recognition can usually

be divided into three stages of processing. First, discriminative points of interest are detected in each image frame. Several choices exist for detecting interest points which include the Harris corner detector [88], discriminative features [183], and scale-invariant features (SIFT) [134]. Next, a descriptor is then computed for each of these chosen feature locations. This descriptor is typically chosen such that it is invariant to local deformations so that pose and lighting changes do not affect these local descriptors significantly. Examples of such popular descriptors that include SIFT [113, 134] or the deformation invariant feature proposed in [128]. Once such descriptors are computed for each feature point, then the object is described using a collection or bag-of-features model [126, 184]. In a bag-of-features model, the geometric relationship between the feature points is usually discarded. Some approaches use a coarse representation of the geometric relationship between feature points to improve discrimination between object categories [66]. The essential advantage of feature-based approaches for object recognition is the fact that because these local features are robust to changes in viewpoint and illumination, they can potentially be robust even under large view changes and occlusions.

4.5.1.1 Feature-Based-Tracking and Recognition Across a Sparse Camera Network

In a smart camera network, local feature-based methods allow for object recognition to be performed simultaneously and independently on each of the smart cameras. Moreover, even when cameras' fields of view do not overlap, such feature-based approaches may be used to maintain the target's identity across the smart camera network. As an example, consider the scenario where a sparse collection of video cameras monitors a large area. Target association across cameras needs to be performed using only the appearance information of these targets since the fields of views of these cameras might not overlap. Unfortunately, global appearance models such as a 2D affine template image are not sufficient because the target's pose will differ when it reappears in another camera view. Moreover, since the targets' 3D structures may differ greatly, it is not possible to use a generic 3D model as in

face tracking. In such a scenario, local feature-based methods in combination with structure from motion techniques provide an effective alternative. In addition, structure from motion-based methods allow for target specific 3D models to be built online as the target moves within the field of view of each camera.

Consider a sparse distribution of cameras (see Figure 4.14) covering a large area with many unobserved regions. Let us suppose a white SUV is seen approaching a camera. Suppose a list of *authorized* vehicles is available with appropriate descriptions, then we could verify if the approaching vehicle is in the authorized list. Verification of vehicle identity across non-overlapping views presents two important challenges: pose and illumination. The models built for each vehicle need to account for possible changes in both.

In [178], we address the problem of establishing identity of vehicles across non-overlapping views, when the vehicle moves on a plane. We use the 3D structure of the vehicle, along with statistical appearance

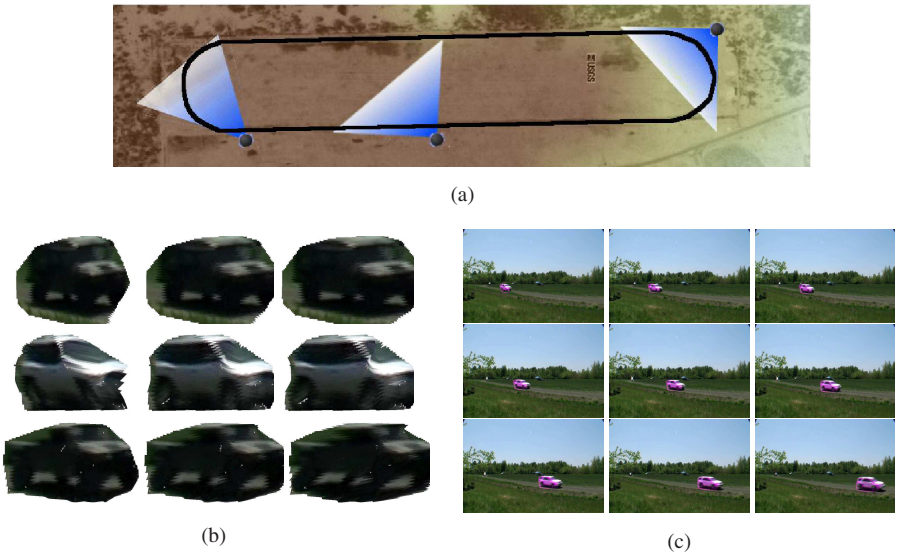


Fig. 4.14 Tracking and verification across non-overlapping views. (a) A schematic top view of the sensing area with fields of view of three cameras shown, (b) 3D structures (with texture maps overlaid) of three vehicles, as estimated from one view. (c) Tracking results with the output inlaid in magenta. These results were first reported in [178].

models as the *fingerprint* representing the vehicles. Estimation of the 3D structure of the vehicle is performed using an approach specifically suited to vehicles exhibiting planar motion [127]. The ability to estimate 3D structure allows us to address the changes in pose. The estimated 3D structure and its texture are used to generate synthetic views of the object. These synthetic views are then compared with the view of the vehicle in other cameras to perform recognition across non-overlapping cameras. We formulate the problem as one of simultaneous tracking and verification [235] (similar to the methods used in Section 4.5) of vehicle identity using the structural and appearance models in the fingerprint. In traditional tracking, the state space contains the position of the object being tracked, while in a simultaneous tracking and verification framework the state space is augmented with the identity of the vehicle. Thus, simultaneous tracking and verification amounts to recursively estimating the position and the identity of the vehicle. Estimating the pose of the object in every frame enables recognition to be performed irrespective of the viewpoint of the camera. Figure 4.14 summarizes results from [178].

5

Statistical Analysis of Structure and Motion Algorithms

5.1 Introduction

An image is the projection of the 3D world onto a 2D image plane. All points on a single line passing through the optical center of the perspective camera project to the same point on the image plane. Therefore, some information about the 3D structure of the object is lost in this projection. To recover and estimate the structure of objects from the captured images, one needs additional information apart from a monocular image. The problem of estimating structure from images can be solved either by using additional information about the objects or by using multiple images. An example of using additional information about the objects in the scene is shape from shading [32]. In shape from shading, the object being imaged is assumed to have Lambertian reflectance and piece-wise constant albedo. This allows us to estimate the structure of the object from the shading exhibited by it. Nevertheless, approaches for estimating structure from a single image are significantly limited both in the robustness of the estimated reconstruction and their applicability. In contrast, using multiple images to estimate the structure of the objects in the scene is both popular and proven to be a robust method for estimating structure.

In order to estimate the structure of the scene using multiple images, three fundamentally different approaches have been used. The first uses a stereo pair of cameras or a multi-camera array. In this scenario, multiple cameras observe the scene, having relative position and orientation known a priori. Point correspondences are estimated between the images acquired by the multiple cameras. These point correspondences allow for triangulation of the original scene point and subsequently, estimation of scene structure. Another common method for estimating structure uses a moving camera. When a moving camera observes a static scene, the various images captured at different instants can be interpreted as being captured from multiple virtual cameras. This solution enables triangulation of scene points just as in the case of structure from stereo. But unlike in stereo, one does not know the position of the cameras and therefore this also needs to be simultaneously estimated. In addition, the continuity and smoothness of the camera motion can be used as additional priors to constrain both the search for point correspondences and for the final structure estimation (see Figure 5.1). These approaches are broadly termed as structure from motion (SfM).

Most approaches for solving the SfM problem can be categorized as follows. Optical flow-based approaches, such as [2, 227], rely on 'inverting' the optical flow equation to estimate 3D structure and 3D motion parameters from the measured image-plane pixel velocities. These approaches usually provide dense estimates of the structure of the scene. Factorization approaches, such as [156, 203, 204], estimate the 3D structure and motion of a sparse set of feature points by expressing the imaging equation in a factored form. These approaches are batch procedures that solve for the unknown structure and motion parameters over several frames simultaneously. Recursive approaches, such as [7, 31], solve for the unknowns sequentially as new data become available. These approaches cast the problem as one of sequential state-estimation. Iterative approaches, such as bundle adjustment [205], cast the problem as a function minimization problem. The cost function to be minimized is the reprojection error. It is minimized by iteratively linearizing the cost function in the neighborhood of the current estimate and applying function minimization techniques, such as the Levenberg–Marquardt algorithm.

Among the various techniques used for SfM, most can be formulated or interpreted as a statistical inference problem with constraints. We briefly discuss statistical methods for structure from motion in this chapter. Several non-statistical algebraic techniques are available for structure from motion, including minimal algorithms such as eight-point algorithm [133] and five-point algorithm [147]. We refer the interested reader to the survey by Oliensis [149] for more details.

In the following we will discuss how the imaging equation studied in Section 2 plays an important role in laying down the fundamental SfM equations. We further illustrate how the SfM problem can be viewed as a statistical estimation problem using two approaches — tracks of feature points and optical flow. In both these approaches, estimating structure is formulated as a dynamical inference problem for which methods studied in Section 3 such as particle filters and Kalman filters can be used. Irrespective of what estimator is used, we discuss how to derive Cramer–Rao lower bounds for SfM that provide a theoretical limit to the best performance achievable.

5.2 Feature-Based Methods

In this section, we discuss the feature-based methods for solving the SfM problem that utilize either image features or optical flow obtained from multiple cameras or a single moving camera. Consider an image sequence with N images. Let us assume that certain feature points are detected and tracked throughout the sequence. Suppose the scene has M features, and their projections in each image are denoted by $x_{i,j} = (u_{i,j}, v_{i,j})^T$ where $i \in \{1, \cdots, M\}$ denotes the feature index and $j \in \{1, \cdots, N\}$ denotes the frame index (see Figure 5.1). For the sake of simplicity, assume that all features are visible in all frames. The SfM problem involves solving for the camera locations and the 3D coordinates of the feature points $\mathbf{X}_{i,j}$ in the world coordinate system. Two paradigms can be considered in this context: batch algorithms and recursive algorithms. Batch algorithms process all the feature points in the *entire* video. This is especially suited for analyzing videos offline, and building detailed models of sites and objects. In contrast, for applications such as robotics and vehicle guidance, the estimation of

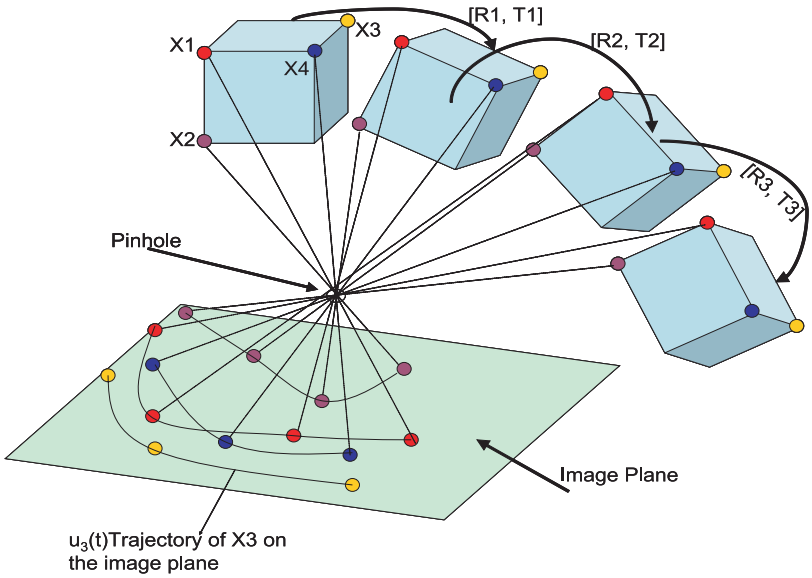


Fig. 5.1 Illustration of the SfM problem: The goal is to estimate the 3D structure of the body undergoing rigid motion from image plane measurements of optical flow or, in this case, trajectories of feature points on the image plane.

structure and motion needs to happen in real time. This motivates the need for recursive algorithms that causally estimate the structure of the scene and its corresponding motion. In recursive algorithms, the traditional approach is to formulate the problem as one of Bayesian inference of a stochastic dynamic system. Such a formulation makes a wide range of inference algorithms available for estimating the parameters of interest (see Section 3). We describe the batch methods first, then the recursive methods.

5.2.1 Batch Methods: Factorization

One of the most elegant approaches for batch processing of feature trajectories to obtain structure and motion parameters is the *factorization algorithm* [203]. Let the camera pose at time instant j ($j = 1, \cdots, N$) be specified by a rotation matrix R_j and a translation vector T_j. The coordinates of a point in the camera system and world system are related by $P_c = R_i P_w + T_i$, where P_c denotes the coordinates of a point in

the camera coordinate system and P_w denotes the coordinates of the same point in the world coordinate system. The 3D coordinates of the world landmarks are denoted by $\underline{X}_i = (X_i, Y_i, Z_i)^T$ for $i = 1, \cdots, M$. We assume an orthographic projection model for the camera. Landmarks are projected onto the image plane according to:

$$\begin{pmatrix} u_{i,j} \\ v_{i,j} \end{pmatrix} = K \cdot \begin{bmatrix} R_j & T_j \end{bmatrix} \underline{X}_i. \tag{5.1}$$

In Equation (5.1), K denotes the 2×3 camera matrix. Let the centroid of the $3D$ points be C, and the centroid of the image projections of all features in each frame be c_j. We can eliminate the translations from these equations by subtracting C from all world point locations, and c_j from the image projections of all features in the j-th frame. Let $\hat{x}_{i,j} = x_{i,j} - c_j$ and $\hat{X}_j = \underline{X}_j - C$. The projection equation can be rewritten as:

$$\hat{x}_{i,j} = P_j \cdot \hat{X}_i. \tag{5.2}$$

In Equation (5.2), $P_j = K \cdot R_j$ denotes the 2×3 projection matrix of the camera. We can stack the image coordinates of all the feature points in all the frames and write it in matrix form as:

$$W = \begin{bmatrix} \hat{x}_{1,1} & \cdots & \hat{x}_{M,1} \\ \vdots & \ddots & \vdots \\ \hat{x}_{1,N} & \cdots & \hat{x}_{M,N} \end{bmatrix} = \begin{bmatrix} P_1 \\ \vdots \\ P_N \end{bmatrix} \begin{bmatrix} \hat{X}_1 & \cdots & \hat{X}_M \end{bmatrix}. \tag{5.3}$$

The matrix W of Equation (5.3) is the measurement matrix, and it has been written as a product of a $2N \times 3$ projection matrix and a $3 \times M$ structure matrix. This factorization implies that the measurement matrix is at most of rank 3. This observation leads to a factorization algorithm to solve for the projection matrices P_j and the $3D$ point locations X_i. The details of the algorithm are described in [203]. Other variants of the factorization approach extend the basic factorization results to para-perspective imaging [156], projective imaging [204], and planar motion [127]. Alternate batch processing algorithms can process feature point trajectories. The most popular of these is the *bundle adjustment* technique [205].

A study of how perturbation errors in 2D image coordinates propagates to the estimates of structure and motion parameters were performed by Sun et al. [198]. They derived first-order perturbation and covariance matrices of the estimates of the 3D shape and motion parameters using matrix perturbation theory. The approach studied how perturbations in the image coordinates effect the three dominant singular values and corresponding eigenvectors of the measurement matrix. It was shown that under small perturbations of image features, the variance of the error in structure and motion estimates grows linearly with the perturbation noise variance. This approximation does not hold for larger perturbation and experiments show that the estimation error explodes with large perturbations of the measurement matrix.

5.2.2 Dynamical System Characterization

In this section, we discuss a recursive solution to the SfM problem by formulating it as a sequential state estimation problem. Formulation as a dynamical system allows for a simple yet powerful description of the evolution of the structure and motion parameters in terms of generative models. As before, these models can be broken down into the state evolution and the observation models.

5.2.2.1 State Space

The state space can be broken down into two parts: the static structure parameters and the dynamic motion parameters [31]. The structure parameters are the 3D locations of the feature points $\mathbf{X}_i, i = 1, \ldots, M$, which are assumed to be static. The motion parameters refer to the camera rotation $R(t)$ and its translation $T(t)$ at a given time instant t. Hence, at time t, the size of the state space is $6t + 3M$.[1] However, the observations for a feature point at a given time instant have two degrees of freedom, making the total number of degrees of freedom in all observations up to at time instant t equal to $2Mt$. The problem is solvable only when $2Mt \geq 6t + 3M$.

[1] There are three degrees of freedom for each structure point, and six degrees of freedom for the rotation + translation of the camera at any given time instant.

The camera rotation can be parameterized in multiple ways [197] including Euler angles, quaternions, and rotation matrices. Euler angles define the overall rotation as a sequence of rotations about three axes, and allow for a unique $3D$ parameterization of almost all rotations. However, the rotation matrix itself is highly non-linear in the Euler angles, involving trigonometric relationships. Quaternions provide a $4D$ global parametrization of the rotation group, where the rotation matrix can be defined using quadratic functions in the quaternions themselves. Finally, rotation matrices can be identified as the space of matrices that satisfy the property $R^T R = I, \det(R) = 1$ or the *special orthogonal group* $SO(n)$. Matrices on this manifold can be generated using exponentials of skew-symmetric matrices.

It is possible to simplify the state space formulation in multiple ways. The first approach assumes that the structure of the scene can be parameterized by only its depth in some fixed camera reference [7]. This approach further assumes that the image plane measurements are noise free, and knowledge of the depth of the scene point in some reference allows us to locate it on its pre-image. Such an assumption reduces the number of unknowns at time t to $6t + M$, making the problem better conditioned.

In the case of recursive SfM, the parameters of interest encompass the motion parameters $(R(t), T(t), t = 1, \ldots, t_0)$ and the structure parameters $(\mathbf{X}_i, i = 1, \ldots, M)$. The observation $\mathbf{Z}$ encompasses the trajectories of all observed feature points, namely $\mathbf{Z} = \{\mathbf{u}_i(t), t = 1, \ldots, t_0, i = 1, \ldots, M\}$. The density $f(\mathbf{Z}|\theta)$ can be written using the basic equations of the dynamical system (see Section 3.4).

5.2.2.2 State Transition Models

The structure parameters $\mathbf{X}_i$ are assumed to be static subject to Brownian evolution. For the motion parameters (rotation $R(t)$ and translation $T(t)$), the most common assumption is to enforce continuity of the motion to various orders. This may include enforcing continuity of the velocity vectors [31] or just that of the motion parameters themselves [160].

5.2.2.3 Observation Models: Reprojection

The most common observation model is that of reprojection of the structure and motion parameters. Let $\mathbf{u}_i(t) \in \mathbb{R}^2$ be the feature point on the image plane associated with the i-th structure point at time t. The reprojection model compares the error between reprojecting the point $\mathbf{X}_i$ under the motion parameters $R(t)$ and $T(t)$ with $\mathbf{u}_i(t)$ as:

$$\mathbf{u}_i(t) = P(K[R(t)\mathbf{X}_i + T(t)]) + \omega_t, \tag{5.4}$$

where ω_t is the observation noise and $P(\cdot)$ the projection operator,

$$P\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \frac{1}{z}\begin{pmatrix} x \\ y \end{pmatrix}, z \neq 0. \tag{5.5}$$

Suitable inference can be performed using an extended Kalman filter (EKF) by linearizing these models [7, 31]. More recent approaches [160] have used particle filters to solve the inference problems. Using particle filters for such high-dimensional problems requires additional modeling assumptions, such as decoupling the structure and motion parameters and solving a set of smaller problems [160]. We briefly discuss these issues next.

5.2.3 Inference using Particle Filters

Particle filters compute the posterior density for non-linear non-Gaussian dynamical systems by approximating the posterior using samples. The accuracy of the approximation depends on the number of the particles used and the skewness of the density. For state spaces that are partially observable, applying particle filters directly may require a prohibitive number of particles for accurate representation of the density. Exploiting additional structure specific to the problem often helps in keeping the inference tractable. Toward this end, Qian and Chellappa [160] propose a particle filter where the motion parameters are recovered independently from feature trajectories. The posterior distribution of the camera motion is used to recover the depth of the feature points using a separate filter for each feature point.

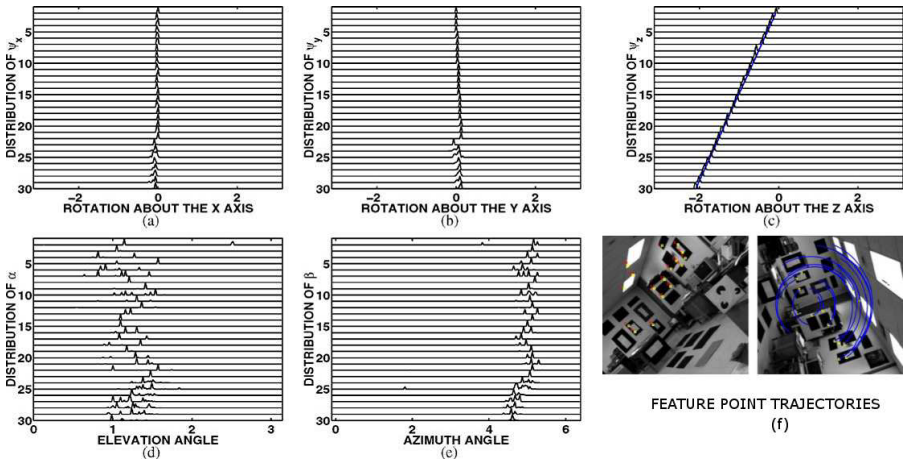


Fig. 5.2 Results on the PUMA2 sequence using particle filters. (a)–(e) show the posterior distributions of the motion parameters, (f) shows feature locations and trajectories. These results were first reported in [160].

The particle filter for obtaining camera motion parameters is formulated with a $5D$ state space comprising camera rotation and translation normalized to avoid the global scale ambiguity. The state transition model is modeled as Brownian motion. Observation model exploits the epipolar constraint induced by the motion parameters, and uses the deviation of a feature point from its epipolar line as the observation error. Structure estimation is done independently for each point by triangulating the trajectory of a feature point using the camera motion posterior as additional input.

The particle filter solution was tested on the 30-frame long PUMA2 sequence (see Figure 5.2). Figure 5.2 shows the motion distributions of this sequence at different time instants as well as the location and trajectories of feature points used to compute the motion of the camera. Plot (c) of Figure 5.2 shows the ground truth (the thick line) and computed distribution of the rotation about the optical axis.

As a second example, a 3D face model is reconstructed from a face sequence. Since the proposed method is feature-based, the depth values of a set of feature points are first computed using the proposed approach and then the depth field of the entire object is obtained by using the

Fig. 5.3 The intensity texture map, tracked feature points on two frames, and images synthesized from the reconstructed 3D model of the face. These results were first reported in [160].

interpolation function griddata in MATLAB 6.1.0. Figure 5.3 shows the intensity map of the face (the up-most-left one) and the reconstructed face model from different viewpoints.

5.2.4 Performance Analysis

In complex problems such as SfM, comparison of the performance of the algorithm against the Cramer–Rao bounds is extremely valuable for an objective evaluation of a particular method. However, as discussed in Section 3.3, application of CRLB to a particular estimation technique has to consider whether the estimator is unbiased. When the estimator is biased, the computation of the lower bound needs to account for the bias. However, such computations require an analytical expression for the bias. Nonetheless the CRLB allows us to characterize the estimation error as a function of variates, such as the number of feature points, number of frames, and observation errors. Toward this objective, CRLB offers a valuable tool in understanding the fundamental limitations and dependencies between the variables influencing the problem and the parameters of interest.

Suppose θ is the parameter of interest, and $\hat{\theta}$ is an estimate, then the CRLB can be stated as:

$$V = \mathrm{E}\left((\hat{\theta} - \theta)(\hat{\theta} - \theta)^T\right) \geq J^{-1}, \tag{5.6}$$

where V is the error covariance of an estimator and J is the Fisher Information matrix. For an unbiased estimation problem, the Fisher Information matrix is given as,

$$J = \mathrm{E}\left(\frac{\partial \log f(\mathbf{Z}|\theta)}{\partial \theta} \frac{\partial \log f(\mathbf{Z}|\theta)}{\partial \theta}^T |\theta\right), \tag{5.7}$$

where $f(\mathbf{Z}|\theta)$ encodes the dependence of the observations $\mathbf{Z}$ on the parameters θ.

Broida and Chellappa [30] consider the special case of SfM when the motion undergone by the camera (or equivalently, the object) is a constant velocity. For this scenario, the motion can be characterized using a $3D$ velocity vector for the translation and a $4D$ skew-symmetric matrix characterizing the angular velocity of the quaternions. In particular, the parameters of interests can be defined using the angular velocity $\omega = (\omega_x, \omega_y, \omega_z)$ and a translational velocity $\mathbf{v} = (\dot{x}, \dot{y}, \dot{z})$. With this, the quaternion $\mathbf{q}(t)$ describing the rotation of the object at time

t can be written as,

$$\mathbf{q}(t) = \left(\frac{\omega_x}{|\omega|} \sin(|\omega|t/2), \frac{\omega_y}{|\omega|} \sin(|\omega|t/2), \frac{\omega_z}{|\omega|} \sin(|\omega|t/2), \cos(|\omega|t/2) \right)^T. \tag{5.8}$$

Further, under the constant velocity model, we can express the motion of an object point X_i in a camera coordinate system as,

$$\mathbf{X}_i(t) = \mathbf{s}_R(t) + R(\mathbf{q}(t))X_i, \tag{5.9}$$

where $R(\mathbf{q}(t))$ is the rotation as defined by the quaternion $\mathbf{q}(t)$, and $\mathbf{s}_R(t)$ is a translating reference point ($\mathbf{s}_R(t) = \mathbf{s}_R(0) + \mathbf{v}t$). Under this assumption, the state transition model is an identity transformation, and the overall probability $f(\mathbf{Z}|\theta)$ depends entirely on the observation models. In this setting, assuming that ω_t in Equation (5.4) is zero-mean Gaussian with covariance $\sigma^2 I$ independent of t,

$$\frac{\partial \log f(\mathbf{Z}|\theta)}{\partial \theta} = \frac{1}{\sigma^2} \sum_{t=1}^{t_0} \left[\sum_{i=1}^{M} \left(\frac{\partial P(K[\mathbf{s}_R(t) + R(\mathbf{q}(t))X_i])}{\partial \theta}^T \right. \right.$$
$$\left. \left. \times \left(P(K[\mathbf{s}_R(t) + R(\mathbf{q}(t))X_i]) - \mathbf{u}_i(t) \right) \right) \right]. \tag{5.10}$$

The Fisher information matrix can be expressed as,

$$J = \frac{1}{\sigma^2} \sum_{t=1}^{t_0} \left[\sum_{i=1}^{M} \left(\frac{\partial P(K[\mathbf{s}_R(t) + R(\mathbf{q}(t))X_i])}{\partial \theta}^T \right. \right.$$
$$\left. \left. \times \frac{\partial P(K[\mathbf{s}_R(t) + R(\mathbf{q}(t))X_i])}{\partial \theta} \right) \right]. \tag{5.11}$$

Since this computation involves Mt_0 sums of rank 2 matrices, it has rank of at most $2Mt_0$. In comparison, the dimensionality of the parameter vector θ is $7 + 3M$. Hence, J is full rank and invertible only when $2Mt_0 \geq 7 + 3M$. Alternately, the parameters are unobservable when $2Mt_0 < 7 + 3M$. Once J is computed, the diagonal of its inverse can be used as a set of lower bounds for the parameters.

All feature-based methods assume that features in the images can be reliably extracted and matched across frames. In many real scenarios,

such as in aerial imagery, few, if any, distinctive features exist. For such situations, flow-based methods are more suitable. The next section will discuss flow-based approaches.

5.3 Flow-Based Methods

Optical flow is the apparent motion of image pixels caused by the actual relative motion between the scene and the camera. Given two images, we are interested in computing the camera motion and structure of the scene using the optical flow information. Let $p(x,y)$ and $q(x,y)$ be the horizontal and vertical components of the flow observed at a point (x,y) in the image. They are related to the 3D object motion and scene depth (under the infinitesimal motion assumption) by,

$$\begin{aligned} p(x,y) &= (-v_x + xv_z)g(x,y) + xy\omega_x - (1+x^2)\omega_y + y\omega_z, \\ q(x,y) &= (-v_y + yv_z)g(x,y) + (1+y^2)\omega_x - xy\omega_y - x\omega_z, \end{aligned} \tag{5.12}$$

where $V = (v_x, v_y, v_z)$ and $\Omega = (\omega_x, \omega_y, \omega_z)$ are the translational and rotational components of the camera motion, $g(x,y) = 1/Z(x,y)$ is the inverse scene depth, and all linear dimensions are normalized in terms of the focal length f of the camera. Rewriting Equation (5.12) as,

$$\begin{aligned} p(x,y) &= (x - x_f)h(x,y) + xy\omega_x - (1+x^2)\omega_y + y\omega_z, \\ q(x,y) &= (y - y_f)h(x,y) + (1+y^2)\omega_x - xy\omega_y - x\omega_z, \end{aligned} \tag{5.13}$$

where $(x_f, y_f) = (v_x/v_z, v_y/v_z)$ is known as the *focus of expansion* (FOE) and $h(x,y) = v_x/Z(x,y)$. We first formulate a two-frame solution for estimating V, Ω and Z from $(p(x,y), q(z,y))$ using Equation (5.13).

5.3.1 Two-Frame Solution

We first consider the case where the FOE is known [192] and then discuss the unknown FOE case. In situations where the FOE does not change appreciably over a few frames, it is possible to estimate the FOE from the first two or three frames and assume that it remains constant for the next few frames. Note that under known FOE, Equation (5.13) is linear in its unknowns, $h(x,y)$ and Ω. We can formulate the flow

relation on all the pixels as follows. Let

$$P = \begin{bmatrix} x(1,1) - x_f & 0 & \cdots & 0 \\ y(1,1) - y_f & 0 & \cdots & 0 \\ 0 & x(1,2) - x_f & \cdots & 0 \\ 0 & y(1,2) - y_f & \cdots & 0 \\ 0 & 0 & \cdots & x(M,M) - x_f \\ 0 & 0 & \cdots & y(M,M) - y_f \end{bmatrix}, \tag{5.14}$$

$$R = \begin{bmatrix} x(1,1)y(1,1) & -(1 + x(1,1)^2) & y(1,1) \\ 1 + y(1,1)^2 & -x(1,1)y(1,1) & -x(1,1) \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ x(M,M)y(M,M) & -(1 + x(M,M)^2) & y(M,M) \\ 1 + y(M,M)^2 & -x(M,M)y(M,M) & -x(M,M) \end{bmatrix}, \tag{5.15}$$

and

$$\mathbf{u} = (p(1,1), q(1,1), \cdots, p(M,M), q(M,M))^T. \tag{5.16}$$

The system of equations relating the flow at each pixel to the motion and structure parameters can be summarized as,

$$\begin{bmatrix} P & R \end{bmatrix} \mathbf{z} = \mathbf{u}, \tag{5.17}$$

where $\mathbf{z} = (h(1,1), \ldots, h(M,M), \omega_x, \omega_y, \omega_z)^T$. Denoting $B = \begin{bmatrix} P & R \end{bmatrix}$, the MMSE solution of $\mathbf{z}$ can be easily obtained by inverting the system of linear equations, $\hat{\mathbf{z}} = B^{-1}(\mathbf{u})$. We can also characterize the uncertainty in the estimate, $\hat{\mathbf{z}}$, when we have a characterization of the covariance of the uncertainty in $\mathbf{u}$. In particular, when the covariance matrix of $\mathbf{u}$ is $R_{\mathbf{u}} = r^2\mathbb{I}$, the covariance of $\hat{\mathbf{z}}$ is given as,

$$R_{\mathbf{z}} = r^2(B^T B)^{-1}. \tag{5.18}$$

When the FOE is unknown, the linear structure of Equation (5.17) is lost. The unknown vector $\mathbf{z}$ is estimated by formulating the reprojection error and minimizing using optimization techniques. Uncertainty in the estimate is computed as follows. Let $\mathbf{z} = \psi(\mathbf{u})$. Expanding ψ in a Taylor series around its mean, $E[\mathbf{u}]$,

$$\psi(\mathbf{u}) = \psi(E[\mathbf{u}]) + D_\psi(E[\mathbf{u}])(u - E[\mathbf{u}]) + O(E[\mathbf{u}])^2, \tag{5.19}$$

where $O(\cdot)$ denotes terms of order 2 or higher and $D_\psi = \frac{\partial \psi}{\partial x}$. Up to a first-order approximation,

$$\psi(\mathbf{u}) - \psi(E[\mathbf{u}]) = D_\psi(E[\mathbf{u}])(u - E[\mathbf{u}]). \tag{5.20}$$

The covariance of $\mathbf{z}$ can then be written as,

$$\begin{aligned} R_\mathbf{z} &= E[(\psi(\mathbf{u}) - \psi(E[\mathbf{u}]))(\psi(\mathbf{u}) - \psi(E[\mathbf{u}]))^T] \\ &= E[D_\psi(E[\mathbf{u}])(u - E[\mathbf{u}])(u - E[\mathbf{u}])^T (D_\psi(E[\mathbf{u}]))^T] \\ &= D_\psi(E[\mathbf{u}]) R_\mathbf{u} D_\psi(E[\mathbf{u}])^T, \end{aligned} \tag{5.21}$$

where $R_\mathbf{u}$ is the covariance matrix of $\mathbf{u}$ and we have made an additional assumption that $E[\mathbf{z}] = \psi(E[\mathbf{u}])$. The exact expression for the covariance $R_\mathbf{z}$ can now be evaluated when the estimation function, ψ, and the covariance matrix $R_\mathbf{u}$ are known. Please see [42] for more details.

5.3.2 Multi-frame Fusion using Stochastic Annealing

Figure 5.4 shows a block-diagram schematic of the complete multi-frame fusion algorithm. The input is a video sequence of a static scene captured by a moving camera. We choose least median square error (LMedS) as our estimate for two-frame depth reconstruction (recall that under unknown FOE, the relation between flow and the unknown motion and structure parameters is non-linear). The two frames may be adjacent ones, or may be farther apart. However, the constraint of small motion in optical flow estimation needs to be borne in mind. Our problem is to design an efficient algorithm to align the depth maps onto a single frame of reference (since the camera is moving), fuse the aligned depth maps in an appropriate way and evaluate the quality of the final reconstruction. All this needs to be done after due consideration is given to the possible sources of error and their effects as outlined earlier.

Roy Chowdhury and Chellappa [42] describe an algorithm for multi-frame fusion. Given pairs of frames, the two-step algorithm described earlier is used to get an LMedS estimate of the motion and depth of the scene points. Covariance of the LMedS is also computed by linearizing the cost function around this estimate. Prior to fusion, the camera motion estimates are used to register all the depth estimates to a common reference. Fusion of the depth estimates is performed using

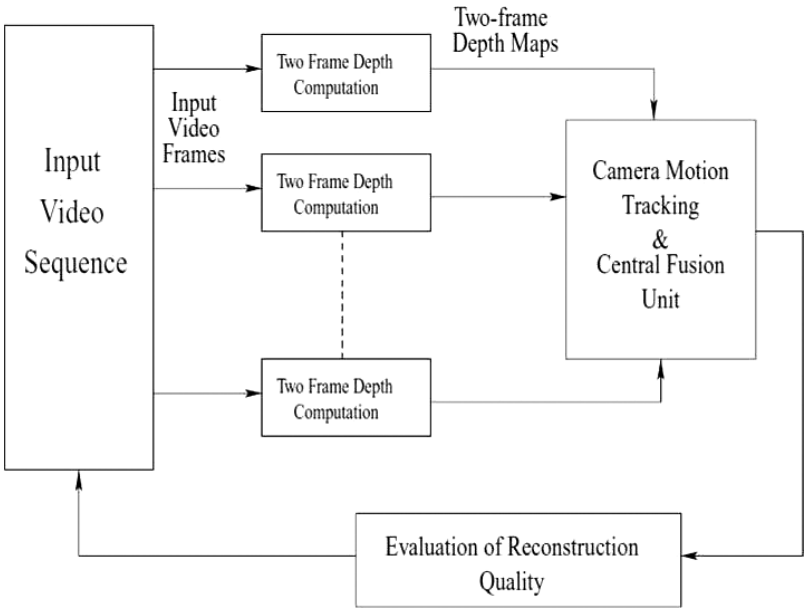


Fig. 5.4 Block diagram of the multi-frame fusion system. This was first reported in [42].

Robbins–Monro stochastic approximation (RMSA) [167]. The reason for choosing stochastic approximation (SA) over other methods is as follows. Stochastic annealing is a recursive estimation method, that chooses an appropriate weight for update, with the guarantee that the limit it converges to is the parameter value sought. It provides an elegant tool to deal with problems where characterization of the error distribution is known. Besides, it provides a recursive algorithm and guarantees local optimality of the estimate, which can be non-linear. On the other hand, the Kalman filter is optimal only among the class of linear filters (in the mean square sense) for any noise distribution. For the Gaussian distribution, it is an optimal filter in the mean square sense. Since LMedS is a non-linear estimator and the distribution of the depth subestimates is unknown, SA is used to obtain an optimal solution based on the method of calculating the quantile of any distribution recursively, following the method originally proposed by Robbins and Monro [167]. The issues of convergence and optimality of RMSA have also been studied in depth.

5.3.3 Experimental Results

In Figure 5.5, we show 3D reconstruction results on a 15-frame video sequence of a human face. The focus of expansion is above the left eye as shown in Figure 5.5(a). Results of 3D reconstruction are shown in Figure 5.5(d)–(f) using the RMSA algorithm described above. To study the effect of bias compensation, we show reconstruction results on public database. This database includes the 3D depth model of faces obtained from a range scanner and a frontal image of the face for texture mapping. We used the 3D model and the texture map to create a sequence of images after specifying the camera motion. The camera motion consisted of translation along the x and z axes and rotation about the y axis. Given this sequence of images, we estimate the 3D model using a 3D face reconstruction algorithm [42] and the bias in the

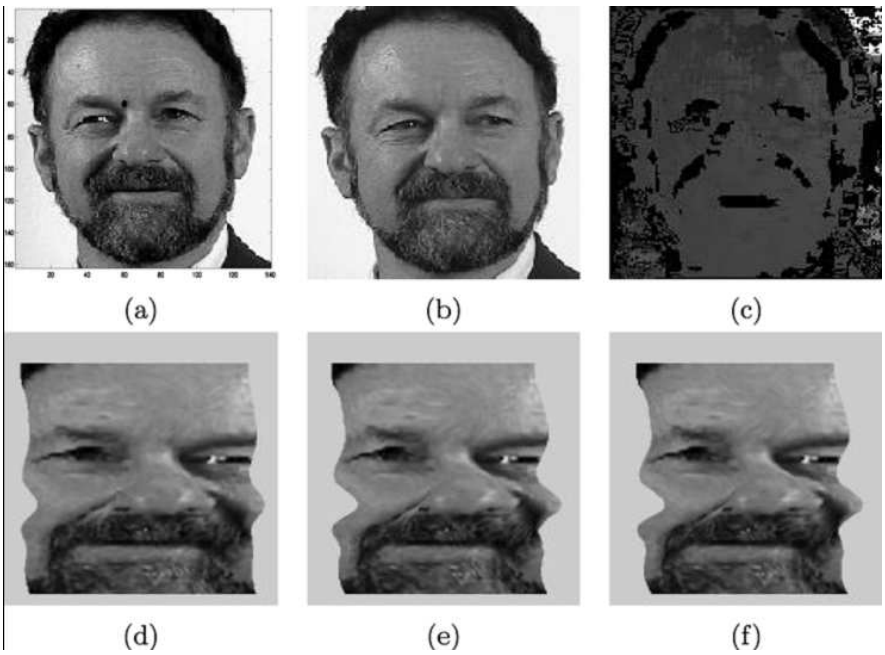


Fig. 5.5 (a) and (b) represent two images from a face video sequence. The focus of expansion is marked on the image in (a) (above the left eye), (c) represents the depth map, with intensity corresponding to depth. The remaining figures (d)–(f) are results of 3D reconstructions from 15 frames for different viewing angles using the RMSA algorithm. These results were first reported in [42].

Table 5.1. Effect of bias on 3D face reconstruction. These results were first reported in [172].

Subject Index	Peak % Bias	Avg. % Error (before bias compensation)	Avg. % Error (after bias compensation)
1 (frame 001)	30	3.8	3.6
2 (frame 002)	34	3.2	3.1
3 (frame 003)	29	3.0	2.7
4 (frame 004)	21	3.9	3.7
5 (frame 005)	26	3.2	3.0

reconstruction. We present here the results on the first five face models in the above-mentioned database. From Table 5.1, we see that the peak value of the bias is a significant percentage of the true depth value. This happens only for a few points. However, it has significant impact on the 3D face model because of interpolation techniques which, invariably, are a part of any method to build 3D models. The third and fourth columns in Table 5.1 represent the root mean square (RMS) error of the reconstruction represented as a percentage of the true depth and calculated before and after bias compensation. The change in the average error after bias compensation is minimal. However, this number is misleading by itself. The average error in reconstruction may be small, but, even one outlier has the potential to create a poor reconstruction. Hence, it is crucial to compensate for the bias in problems related to 3D reconstruction from a monocular video.

5.3.4 Performance Analysis

We analyze the statistical ambiguities inherent to the optical flow-based SfM approach. Given the random nature of the noisy image measurements, any motion and structure estimator will have an associated uncertainty. We now derive the CRLB for the SfM problem using optical flow. This was first reported by Young and Chellappa [227]. In particular, consider the case when the measurements are contaminated by independent and identically distributed (i.i.d.) additive Gaussian noise, assume the measurement model:

$$\mathbf{u} = B(\mathbf{z}) + \underline{\mathbf{n}}, \tag{5.22}$$

where B is deterministic, $\mathbf{z} = (x_f, y_f, \omega_x, \omega_y, \omega_z, Z(x,y))$ and the components of the noise vector $\underline{\mathbf{n}}$ are i.i.d. Gaussian random variables with mean zero and variance σ^2. Then the Fisher information matrix simplifies to:

$$\begin{aligned} J &= \frac{1}{\sigma^2}\left(\frac{\partial B}{\partial \mathbf{z}}\right)^T\left(\frac{\partial B}{\partial \mathbf{z}}\right) \\ &= \frac{1}{\sigma^2}\sum_i \left(\frac{\partial B_i}{\partial \mathbf{z}}\right)^T\left(\frac{\partial B_i}{\partial \mathbf{z}}\right), \end{aligned} \tag{5.23}$$

in which B_i is the i-th component of B. If the optical flow field is available at M image points, the free parameters are the five motion parameters,

$$\underline{\kappa} \equiv (x_f, y_f, \omega_x, \omega_y, \omega_z)^T,$$

and the M structure parameters,

$$\underline{Z} \equiv (Z_1, Z_2, Z_3, \ldots\ldots, Z_M)^T.$$

The measurements are:

$$u_i, v_i, i = 1, 2, \ldots, M.$$

such that, assume that the noise in the extracted motion field obeys the simple Gaussian model,

$$\begin{aligned} u_i &= B_{ui}(x_f, y_f, \omega_x, \omega_y, \omega_z, Z_i) + \mathbf{n_{ui}}, \\ v_i &= B_{vi}(x_f, y_f, \omega_x, \omega_y, \omega_z, Z_i) + \mathbf{n_{vi}}, \end{aligned} \tag{5.24}$$

where $\mathbf{n_{ui}}$ and $\mathbf{n_{vi}}$ are i.i.d. zero-mean Gaussian random variables with variance σ^2. Quantitatively, the CRLB can be used to investigate the variances of the parameter estimates. Using Equation (5.23) with the simple model of Equation (5.24), the Fisher information matrix is given as,

$$J = \frac{1}{\sigma^2}\sum_{i=1}^{M}\left[\left(\frac{\partial B_{ui}}{\partial \mathbf{z}}\right)^T\left(\frac{\partial B_{ui}}{\partial \mathbf{z}}\right) + \left(\frac{\partial B_{vi}}{\partial \mathbf{z}}\right)^T\left(\frac{\partial B_{vi}}{\partial \mathbf{z}}\right)\right], \tag{5.25}$$

where

$$\begin{aligned}\mathbf{z} &= (x_f, y_f, \omega_x, \omega_y, \omega_z, Z_1, Z_2, ..., Z_M)^T \\ &= \left(\underline{\kappa}^T, \underline{Z}^T\right)^T.\end{aligned}$$

It has the partition

$$J = \begin{pmatrix} \alpha & \beta \\ \beta^T & \gamma \end{pmatrix}, \tag{5.26}$$

where α is a 5×5 matrix, and

$$\begin{aligned}\alpha &= \frac{1}{\sigma^2}\sum_{i=1}^{M}\left[\left(\frac{\partial B_{ui}}{\partial \underline{\kappa}}\right)^T\left(\frac{\partial B_{ui}}{\partial \underline{\kappa}}\right) + \left(\frac{\partial B_{vi}}{\partial \underline{\kappa}}\right)^T\left(\frac{\partial B_{vi}}{\partial \underline{\kappa}}\right)\right], \\ \beta &= \frac{1}{\sigma^2}\sum_{i=1}^{M}\left[\left(\frac{\partial B_{ui}}{\partial \underline{\kappa}}\right)^T\left(\frac{\partial B_{ui}}{\partial \underline{Z}}\right) + \left(\frac{\partial B_{vi}}{\partial \underline{\kappa}}\right)^T\left(\frac{\partial B_{vi}}{\partial \underline{Z}}\right)\right], \\ \gamma &= \frac{1}{\sigma^2}\sum_{i=1}^{M}\left[\left(\frac{\partial B_{ui}}{\partial \underline{Z}}\right)^T\left(\frac{\partial B_{ui}}{\partial \underline{Z}}\right) + \left(\frac{\partial B_{vi}}{\partial \underline{Z}}\right)^T\left(\frac{\partial B_{vi}}{\partial \underline{Z}}\right)\right].\end{aligned} \tag{5.27}$$

From Equation (5.12), we note that γ is a diagonal matrix with the j-th diagonal term,

$$[\gamma]_{jj} = \frac{1}{\sigma^2}\left[(x_j - x_f)^2 + (y_j - y_f)^2\right]. \tag{5.28}$$

Thus, γ^{-1} is also diagonal with the j-th term,

$$\left[\gamma^{-1}\right]_{jj} = \frac{\sigma^2}{(x_j - x_f)^2 + (y_j - y_f)^2}. \tag{5.29}$$

Since the CRLBs for the motion parameters $(x_f, y_f, \omega_x, \omega_y, \omega_z)$ are the first five diagonal terms of J^{-1}, they are the five diagonal terms of the 5×5 matrix Q defined as,

$$J^{-1} = \begin{pmatrix} \alpha & \beta \\ \beta^T & \gamma \end{pmatrix}^{-1} \equiv \begin{pmatrix} Q & S \\ S^T & G \end{pmatrix}. \tag{5.30}$$

When J is nonsingular, it can be shown that [189]:

$$\begin{aligned} Q &= (\alpha - \beta\gamma^{-1}\beta^T)^{-1} = \alpha^{-1} + \alpha^{-1}\beta G\beta^T\alpha^{-1}, \\ G &= (\gamma - \beta^T\gamma^{-1}\beta)^{-1} = \gamma^{-1} + \gamma^{-1}\beta^T Q\beta\gamma^{-1}, \\ S &= -Q\beta\gamma^{-1} = -\alpha^{-1}\beta G. \end{aligned} \tag{5.31}$$

By Equations (5.27), (5.29), and (5.32), Q can be calculated as,

$$\begin{aligned} Q &= (\alpha - \beta\gamma^{-1}\beta^T)^{-1} \\ &= \sigma^2 \left\{ \sum_{j=1}^{M} \frac{\begin{array}{c}\left[(y_j - y_f)\frac{\partial B_{uj}}{\partial \underline{\kappa}} - (x_j - x_f)\frac{\partial B_{vj}}{\partial \underline{\kappa}}\right]^T \\ \times \left[(y_j - y_f)\frac{\partial B_{uj}}{\partial \underline{\kappa}} - (x_j - x_f)\frac{\partial B_{vj}}{\partial \underline{\kappa}}\right]\end{array}}{(x_j - x_f)^2 + (y_j - y_f)^2} \right\}^{-1} \end{aligned} \quad .(5.32)$$

Since the CRLBs of the five motion parameters $(x_f, y_f, \omega_x, \omega_y, \omega_z)$ are the diagonal terms of Q, they can be computed for any 3D motion and surface by inverting a 5×5 matrix. From Equation (5.32), it can also be seen that the CRLBs are independent of the rotation parameters (A, B, C) and proportional to σ^2. As large CRLBs indicate high uncertainty in the estimates of the motion parameters, we have the following observations for the ambiguity of the parameters of a general motion in an arbitrary environment:

- the motion parameters are ambiguous if the noise level of the optical flow field is high; and
- the ambiguity of the motion parameters (both translational and rotational parameters) is independent of the true rotational motion.

6

Shape, Identity, and Activity Recognition

6.1 Introduction

The analysis and interpretation of video data comprises an important part of modern vision systems. Applications of video-based systems arise in areas such as biometrics, surveillance, motion-synthesis, and Web-based user interfaces. A common requirement among these different applications is to learn statistical models of appearance and motion from a collection of images and videos, then use them for recognition. Several image analysis tasks, such as object recognition, activity analysis, and gait-based human identification, amount to comparing the shapes of objects and characterizing the variation of shape with time. It has been shown that the shape of the outer contour of objects can be used as an efficient descriptor of the object. Similarly, several video-processing tasks reduce to comparing shape sequences. The essential information conveyed by the video can usually be captured by analyzing the boundary of each object as it changes with time.

There are several situations where we are interested in studying either the shape of an object or the way in which the shape of an object changes with time. The manner in which this shape changes

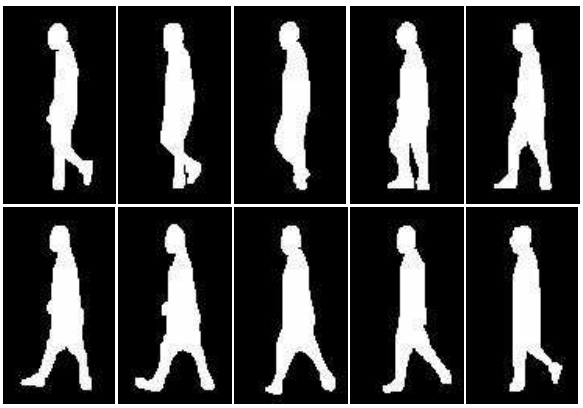


Fig. 6.1 Sequence of shapes as a person walks frontoparallely. It is clear that these belong to a walking individual. This shows that shape is a good feature for action recognition and gait analysis. Image courtesy [217]. Original video sequences were collected at University of Southern Florida as part of the DARPA HID effort [154].

provides cues about the nature of the object and sometimes even about the activity performed by the object. Consider the images shown in Figure 6.1. It is fairly easy to recognize that these represent the silhouette of a walking human. Apart from providing information about the activity being performed, often the manner of shape change provides valuable insights into the identity of the object.

As a motivating example, we shall consider the problem of gait-based recognition. Gait is defined as the style or manner of walking. Studies in psychophysics suggest that people can identify familiar individuals using just their gait. This has led to a number of automated vision-based algorithms that use gait as a biometric. Using gait as a biometric has the advantage of being non-intrusive, meaning it does not require co-operation from subjects and performs reliably at moderate to large distances from the subjects. Several experiments on large, publicly available databases [180] have strongly indicated that the discriminative information in gait is present in the silhouettes of the subjects and the dynamics of the silhouette during a complete walking cycle. The first row in Figure 6.2 shows the tracked bounding box around the object in each video-frame. Since the subject is moving, tracking needs to be performed before such a bounding box can be

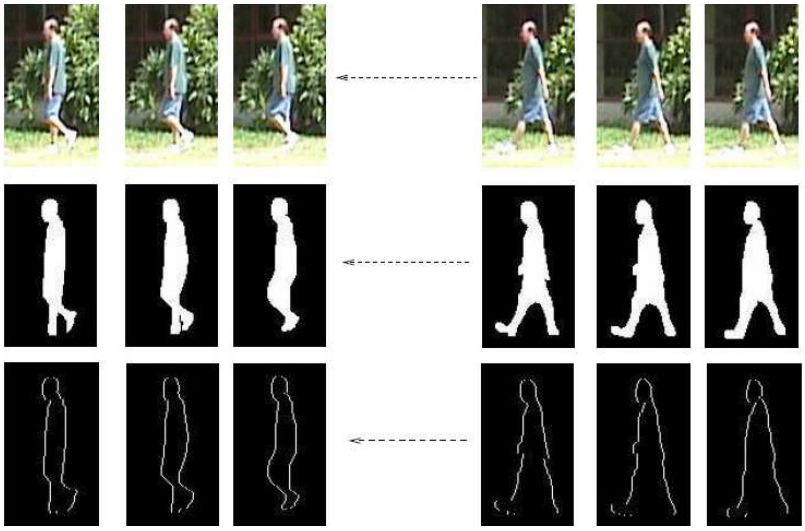


Fig. 6.2 (Top row) Image within the tracked bounding box as a subject is walking. (Middle row) Binary background subtracted image sequence. (Bottom row) Extracted shape sequence containing the discriminative information for classification. Image courtesy [217].

extracted. Background subtraction from each camera view provides us the binary image shown in the second row. The third row shows the extracted silhouette. Note that the characteristics of the individual's shape and identity and the subject's unique mannerisms during walking are captured in the sequence of shapes shown in the third row of Figure 6.2.

Gait-based person identification differs from most other traditional biometrics in the sense that some of the discriminative information is present in the dynamics or the temporal variation in the extracted features. Therefore, it becomes important to either model the dynamics explicitly or to account for the changes in dynamics by performing explicit time normalization. Some popular methods that model and account for both the variations in shape and the variations in dynamics during gait are the hidden Markov models (HMMs) [106, 132], dynamic time warping (DTW) [214, 217], and time series methods like the autoregressive moving average (ARMA) model [217]. HMMs are discussed in greater detail in Section 3.4.1. During training, the model parameters (HMM, ARMA, etc.) for each person in the gallery

are stored for all available views. During the test phase, the model parameters for the test video are estimated and compared with all the model parameters stored in the gallery in order to perform recognition using MAP classification. As this example illustrates, shape analysis and shape sequence analysis are crucial to several problems, including human pose estimation, gait-based person identification, human activity recognition, shape-based object recognition, etc. In this section, we will study the problem of shape and appearance-based activity recognition, gait analysis, and human recognition. By imposing parametric and non-parametric models for the evolution of features, we will discuss how maximum likelihood and MAP estimates of identity or activity can be derived.

6.2 Shape Representations

An object's shape can be defined and parameterized in several ways. In this section we discuss several standard methods for representing shapes. Most of the approaches described below extract descriptors derived from the locations of distinctive feature positions — such as the boundary of an object, corners of an object, etc. Pavlidis [152] categorized shape descriptors as taxonomies according to different criteria. Descriptors that use the points on the boundary of the shape are called external (or boundary) descriptors [14, 110, 153], while those that describe the interior of the object are called internal (or global) descriptors [26, 93]. Descriptors that represent shape as a scalar or as a feature vector are called numeric, while those, like the medial axis transform, that describes the shape as another image are called non-numeric descriptors. Descriptors are also classified as information preserving or not depending on whether the descriptor allows accurate shape reconstruction.

6.2.1 Global Methods for Shape Matching

Global shape matching procedures treat the object as a whole and describe it using features extracted from the object. These methods have the disadvantage that they assume the given image is segmented into various objects, which is itself not an easy problem. In general,

these methods cannot handle occlusion and are not particularly robust to segmentation errors. Popular moment-based descriptors of the object, such as [39, 93, 117], are global and numeric descriptors. Goshtasby [81] used the pixel values corresponding to polar coordinates centered around the shape's center of mass, the shape matrix, as a description of the shape. Parui et al. [150] used relative areas occupied by the object in concentric rings around the centroid of the objects as a description of the shape. Blum and Nagel [26] used the medial axis transform to represent the shape.

6.2.2 Boundary Methods for Shape Matching

Shape matching methods based on an object's boundary or on a set of pre-defined landmarks have the advantage that they can be represented using a 1D function. In the early 1960s, Freeman [73] used chain coding (a method for coding line drawings) for the description of shapes. Arkin et al. [6] used the turning function for comparing polygonal shapes. Persoon and Fu [153] described the boundary as a complex function of the arc-length. Kashyap and Chellappa [110] used a circular autoregressive model of the distance from the centroid to the boundary to describe the shape. The problem with the Fourier representation [153] and the autoregressive representation [110] is that the local information is lost in these methods. Srivastava et al. [195] propose differential geometric representations of continuous planar shapes.

Several authors have recently described shape as a set of finite ordered landmarks. Kendall [114] provided a mathematical theory for the description of landmark-based shapes. Bookstein [29] and Dryden and Mardia [58] have furthered the understanding of such landmark-based shape descriptions. A lot of work has been accomplished on planar shapes [75, 158]. Prentice and Mardia [158] provided a statistical analysis of shapes formed by matched pairs of landmarks on the plane. They provided inference procedures on the complex plane and a measure of shape change in the plane. Berthilsson [15] and Dryden [57] describe a statistical theory for shape spaces. Projective shape and their respective invariants are discussed in [15], while shape models, metrics, and their role in high level vision are discussed in [57].

The shape context [14] of a particular point in a point set captures the distribution of the other points with respect to it. Belongie et al. [14] use the shape context for the problem of object recognition. The softassign Procrustes matching algorithm [165] simultaneously establishes correspondences and determines the Procrustes fit.

6.3 Manifold Representation of Shapes

In this section, we will review methods based on describing a shape using a finite set of landmark points. In applications involving humans and their activities, the silhouette of the human body contains distinctive landmarks such as the hands, legs, and torso. This suggests the use of a representation that exploits the entire information offered by the location of landmarks instead of relying on coarse features. Since the descriptor for a shape needs to be invariant to a variety of transformations (translation, rotation, and scale), these descriptors do not usually live on Euclidean spaces. Under the requirement of rotation, scale, and translation invariance, the shape space can be analytically represented as a complex spherical manifold. Under the requirement of full affine invariance (rotation, non-uniform scale, skew, translation), the shape space can be analytically described as a Grassmann manifold. This choice then leads to appropriate distance measures on the shape space.

To develop accurate inference algorithms on the manifolds discussed above we need to (a) understand the geometric structure of these manifolds, (b) derive appropriate distance measures, and (c) develop probability distribution functions (pdf) and estimation techniques consistent with the geometric structure of these manifolds. Given a database of examples and a query, the following two questions are usually addressed — (a) what is the 'closest' example to the query in the database?, (b) what is the 'most probable' class to which the query belongs? A systematic solution to these problems involves a study of the manifold on which the data lie. The answer to the first question involves study of the geometric properties of the manifold, which then leads to appropriate definitions of distances on the manifold (geodesics). The answer to the second question involves statistical modeling of inter- and

intra-class variations on the manifold, and extends far beyond simply defining distances. Given several samples per class, one can derive efficient probabilistic models on the manifold by exploiting the class ensemble populations that are also consistent with the geometric structure of these manifolds. This offers significant improvements in performance compared to a distance-based nearest-neighbor classifier. In addition, statistical models also provide generative capabilities via appropriate sampling strategies.

6.3.1 Kendall's Statistical Shape and the Spherical Manifold

'Shape is all the geometric information that remains when location, scale and rotational effects are filtered out from the object.' Using this definition, Kendall [58] provides a description of the various tools in statistical shape analysis. Kendall's statistical shape is a sparse descriptor of the shape that describes the configuration of m landmark points in an k-dimensional space as a $m \times k$ matrix containing the coordinates of the landmarks. In our analysis we have a $k = 2$ dimensional space, so it is convenient to describe the shape vector as an m-dimensional complex vector.

Let the configuration of a set of m landmark points be given by an m-dimensional complex vector containing the positions of landmarks. Let us denote this configuration as X. Centered pre-shape is obtained by subtracting the mean from the configuration and then scaling to norm one. The centered pre-shape is given by:

$$Z_c = \frac{CX}{\| CX \|}, \quad where \quad C = I_m - \frac{1}{m} 1_m 1_m^T, \tag{6.1}$$

where I_m is an $m \times m$ identity matrix and 1_m is an m-dimensional vector of ones.

The pre-shape vector extracted by the method described above lies on a complex spherical manifold, i.e., $\|Z_c\| = 1$. Therefore, a concept of distance between two shapes must include the non-Euclidean nature of the shape space. Several distance measures have been defined in [58]. Consider two complex configurations X and Y with corresponding pre-shapes α and β. The full Procrustes distance between the configurations X and Y is defined as the Euclidean distance between the full

Procrustes fit of α and β. The full Procrustes fit is chosen to minimize:

$$d(Y,X) = \| \beta - \alpha s e^{j\theta} - (a + jb)1_k \|, \tag{6.2}$$

where s is a scale, θ is the rotation, and $(a + jb)$ is the translation. The full Procrustes distance is the minimum full Procrustes fit, i.e.,

$$d_{Full}(Y,X) = \inf_{s,\theta,a,b} d(Y,X). \tag{6.3}$$

We note that the pre-shapes are actually obtained after filtering the effects of translation and scale. Hence, the translation value that minimizes the full Procrustes fit is given by $(a + jb) = 0$, while the scale $s = |\alpha^* \beta|$ is unity. The rotation angle θ that minimizes the full Procrustes fit is given by $\theta = arg(|\alpha^* \beta|)$. The partial Procrustes distance between configurations X and Y is obtained by matching their respective pre-shapes α and β as closely as possible over rotations, but not scale. So,

$$d_{Partial}(X,Y) = \inf_{\Gamma \epsilon SO(m)} \| \beta - \alpha\Gamma \| . \tag{6.4}$$

It is interesting to note that the optimal rotation θ remains the same whether we compute the full Procrustes distance or the partial Procrustes distance. The Procrustes distance $\rho(X,Y)$ is the closest great circle distance between α and β on the pre-shape sphere. The minimization is performed over all rotations. Thus, ρ is the smallest angle between complex vectors α and β over rotations of α and β. The three distance measures defined above are all trigonometrically related as:

$$d_{Full}(X,Y) = \sin\rho, \tag{6.5}$$

$$d_{Partial}(X,Y) = 2\sin\left(\frac{\rho}{2}\right). \tag{6.6}$$

When the shapes are very close to each other, little difference exists between the various shape distances.

6.3.2 The Tangent Space

The shape tangent space is a linearization of the spherical shape space around a particular pole. Figure 6.3 illustrates that around a local neighborhood of a point, which is denoted as the pole, there exists

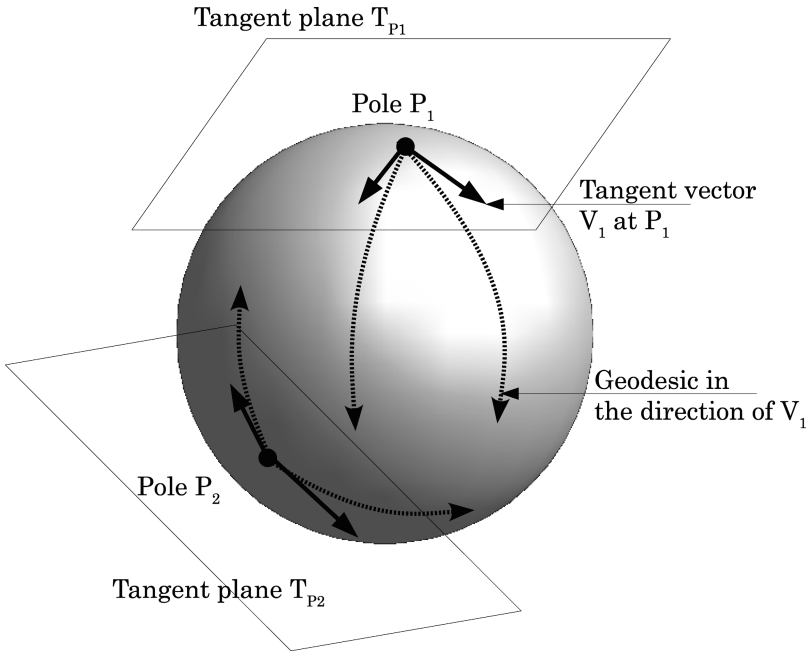


Fig. 6.3 Tangent plane at a pole of the manifold is a local linear approximation of the manifold. The mapping from the tangent plane to the manifold is called the exponential map, and its inverse is called the logarithmic map.

a unique local mapping that takes points from the tangent space to the manifold and vice versa. The mapping from the tangent space to the manifold is called the exponential map, and its inverse is called the inverse-exponential or logarithmic map. Usually, the Procrustes mean shape of a set of similar shapes (Y_i) is chosen as the pole for the tangent space coordinates. The Procrustes mean shape (μ) is obtained by minimizing the sum-of-squares of full Procrustes distances from each shape Y_i to the mean shape, i.e.,

$$\mu = \arg \inf_{\mu} \sum_{i} d^2_{Full}(Y_i, \mu). \tag{6.7}$$

The pre-shape formed by m points lies on an $m-1$ dimensional complex hypersphere of unit radius. If the various shapes in the data are close to each other, then these points on the hypersphere will also lie near one another. The Procrustes mean of this data set will also lie close

to these points. Therefore, the tangent space constructed with the Procrustes mean shape as the pole is an approximate linear space for these data. The Euclidean distance in this tangent space provides a good approximation to various Procrustes distances d_{Full}, $d_{Partial}$, and ρ in shape space in the vicinity of the pole. The tangent space has the advantage of being Euclidean, enabling the use of standard statistical methods for modeling and classification. The Procrustes tangent coordinates of a pre-shape α are given by:

$$v(\alpha, \mu) = \alpha\alpha^*\mu - \mu|\alpha^*\mu|^2. \tag{6.8}$$

where μ is the Procrustes mean shape of the data.

6.3.3 Affine Shape Space and the Grassmann Manifold

The shape observed in an image is a perspective projection of the original shape. To account for this, shape theory studies the equivalent class of all configurations obtained by a specific class of transformation (e.g., linear, affine, projective) on a single basis shape. Suppose, a shape is represented by a set of landmark points on its contour, represented by an $m \times 2$ matrix $S = [(x_1, y_1); (x_2, y_2); \ldots; (x_m, y_m)]$, of the set of m landmarks of the centered shape. The *shape space* of a base shape is the set of equivalent configurations obtained by transforming the base shape by an appropriate spatial transformation. For example, the set of all affine transformations of a base shape forms its *affine shape space*. More rigorously, let $\chi = (x_1, x_2, \ldots, x_m)$ be a configuration of m points where each $x_i \in \mathbb{R}^2$. Let γ be a transformation on $\mathbb{R}^2$. For example, γ could belong to the affine group, linear group, or projective group. Let

$$A(\gamma, (x_1, \ldots, x_m)) = (\gamma(x_1), \ldots, \gamma(x_m)) \tag{6.9}$$

be the *action* of γ on the point configuration.

In particular, the *affine shape space* [79, 190] is important because the effect of the camera location and orientation can be approximated as affine transformations on the original base shape. The affine transforms of the shape can be derived from the base shape simply by multiplying the shape matrix S by a 2×2 full-rank matrix on the right (translations are removed by centering). Multiplication by a full-rank

matrix on the right preserves the column space of the matrix S. Linear subspaces such as these can be identified as points on a Grassmann manifold [151]. Thus, all affine deformations of the same base shape map to the same point on the Grassmann manifold [190], and a systematic study of affine shape space requires a study of the Grassmann manifold. We next discuss distance measures and statistical models on the Grassmann manifold. Specifically, parametric and non-parametric distributions and parameter estimation are reviewed. Procrustes analysis and corresponding distance measures on the manifolds are also presented in [41].

6.3.3.1 Definitions

The Grassmann Manifold $G_{k,m}$ [41]: The Grassmann manifold $G_{k,m}$ is the space whose points are *k-planes* or k-dimensional hyperplanes (containing the origin) in $\mathbb{R}^m$.

Points on the Grassmann manifold are usually represented by tall-thin orthonormal matrices Y such that the columns of Y span the corresponding subspace. Since several orthonormal bases exist for a given subspace, each point on the Grassmann manifold can be interpreted as an equivalence class over the set of orthonormal basis. More formally, the set of $m \times k$ orthonormal matrices is defined as the Stiefel manifold. The Grassmann manifold is then defined in terms of equivalence classes on the Stiefel manifold.

The Stiefel Manifold $V_{k,m}$ [41]: The Stiefel manifold $V_{k,m}$ is the space whose points are *k-frames* in $\mathbb{R}^m$, where a set of k orthonormal vectors in $\mathbb{R}^m$ is called a k-frame in $\mathbb{R}^m$ ($k \leq m$). Each point on the Stiefel manifold $V_{k,m}$ can be represented as an $m \times k$ matrix X such that $X^T X = I_k$, where I_k is the $k \times k$ identity matrix.

An equivalent definition of the Grassmann manifold is as follows: to each k-plane ν in $G_{k,m}$ corresponds a unique $m \times m$ orthogonal projection matrix P idempotent of rank k onto ν. If the columns of an $m \times k$ orthonormal matrix Y span ν, then, $YY^T = P$.

For the case of $k = 1$, the Stiefel manifold reduces to the unit hypersphere in m-dimensions. Each point on the manifold represents a vector of unit length. Similarly, for $k = 1$ the Grassmann manifold reduces to

the real projective space, which consists of all lines passing through the origin.

6.3.3.2 Distances on the Grassmann Manifold

The Stiefel and Grassmann manifolds are endowed with a Riemannian structure that lends itself to computation of distances between points on the manifold via geodesics [1, 59]. Instead of geodesic computations, we adopt the Procrustes distance proposed in [41] that is defined in the ambient Euclidean space. As discussed below, this choice results in efficient computation of distances and the class conditional probability density estimators on the manifolds.

Procrustes representations and corresponding distances are defined to be invariant to specific classes of transformations. Assuming that each point on the Grassmann manifold is represented by a representative element from the equivalence class on the Stiefel manifold, the transformation to which we seek invariance is within plane rotations of the basis vectors. This leads us to the representation as defined below [41].

Procrustes Representation: A point X on $G_{k,m}$ is identified with an equivalence class of $m \times k$ matrices XR in $V_{m,k}$, for orthogonal R. The squared Procrustes distance for two given matrices, X_1 and X_2 on $G_{k,m}$, is the smallest squared Euclidean distance between any pair of matrices in the corresponding equivalence classes. Hence

$$d^2_{procrust}(X_1, X_2) = \min_R tr(X_1 - X_2R)^T(X_1 - X_2R) \tag{6.10}$$

$$= \min_R tr(R^TR - 2X_1^TX_2R + I_k). \tag{6.11}$$

Lemma: Let A be a $k \times k$ constant matrix. Consider the minimization of the quadratic function $g(R) = tr(R^TR - 2A^TR)$ of a matrix argument R.

(1) If R varies over the space $R_{k,k}$ of all $k \times k$ matrices, the minimum is attained at $R = A$.
(2) If R varies over the space of all $k \times k$ positive semi-definite matrices, the minimum is attained at $R = B^+$, where B^+ is the positive semi-definite part of $B = \frac{1}{2}(A + A^T)$.

(3) If R varies over the orthogonal group $O(k)$, the minimum is attained at $R = H_1 H_2^T = A(A^T A)^{-1/2}$, where $A = H_1 D H_2^T$ is the singular value decomposition of A.

We refer the reader to [41] for proofs. Thus, for the case of the first constraint, where R varies over the space $R_{k,k}$ of all $k \times k$ matrices, the distance is given by $d^2_{procrust}(X_1, X_2) = tr(I_k - A^T A)$, where $A = X_1^T X_2$. We have used this metric in all experiments reported in this paper. A closer inspection reveals that these measures are 'divergences' because they are not symmetric in their arguments, hence are not true distance functions. This can be solved by defining a new metric as the average of the divergence between the two points taken in forward and backward directions.

Note that the Procrustes representation defines an equivalence class of points on the Stiefel manifold which are related by a *right* transformation. This directly relates to the interpretation of the Grassmann manifold as the orbit-space of the Stiefel manifold. All points on the Stiefel manifold related by a right transformation map to a single point on the Grassmann manifold. To compare two subspaces represented by two orthonormal matrices, say X_1 and X_2, we compute their Procrustes distance on the Stiefel manifold. We do not explicitly use the representation of points on the Grassmann manifold as $m \times m$ idempotent projection matrices (Section 6.3.3.1). Instead, the Procrustes representation leads to methods that are more computationally efficient, as opposed to working with large $m \times m$ matrices.

6.3.3.3 Non-parametric Kernel Density Estimation

Parametric density forms can provide analytical tractability for estimation problems, but imposing parametric density forms on real-world data can be inappropriate. Kernel methods for estimating probability densities have proved popular in several pattern recognition problems in recent years [21], driven by improvements in computational power. Kernel methods provide a better fit to the available data than simpler parametric forms. Given several examples from a class $(X_1, X_2, \ldots, X_n)$ on the manifold $V_{k,m}$, the class conditional density can be estimated using an appropriate kernel function. We first assume an appropriate choice

of a divergence or distance measure on the manifold (Section 6.3.3.2). For the Procrustes distance, $d^2_{procrust}$, the density estimate is given by [41] as:

$$\hat{f}(X;M) = \frac{1}{n}C(M)\sum_{i=1}^{n} K[M^{-1/2}(I_k - X_i^T X X^T X_i)M^{-1/2}], \quad (6.12)$$

where $K(T)$ is the kernel function, M is a $k \times k$ positive definite matrix which plays the role of the kernel width or a smoothing parameter. $C(M)$ is a normalizing factor chosen so the estimated density integrates to unity. The matrix-valued kernel function $K(T)$ can be chosen in several ways. Most of the results shown in this chapter were originally presented in [207] and used $K(T) = \exp(-tr(T))$.

6.4 Comparing Sequences on Manifolds

Consider a situation having two shape sequences, and we wish to compare their similarities. Since, these shape sequences may differ in length (number of frames), we need to perform time normalization (scaling). Dynamic time warping (DTW), which has been successfully used by the speech recognition [164] community is an ideal candidate for performing this non-linear time normalization. However, certain modifications to the original DTW algorithm are also necessary to account for the non-Euclidean structure of the shape space, i.e., we should be able to incorporate a distance consistent with the geometry of the space under consideration (complex spherical/Grassmann).

6.4.1 Dynamic Time Warping

Dynamic time warping is a method for computing a non-linear time normalization between a template vector sequence and a test vector sequence. These two sequences could have differing lengths. Forner–Cordero et al. [70] show experiments that indicate that the intra-personal variations in gait of a single individual can be better captured by DTW rather than by linear warping. The DTW algorithm is based on dynamic programming and computes the best non-linear time normalization of the test sequence in order to match the template sequence, by performing a search over the space of all allowed time

normalizations. This space is constructed using certain temporal consistency constraints. We list the temporal consistency constraints in our implementation.

- End point constraints: The beginning and the end of each sequence are rigidly fixed. For example, if the template sequence is of length N and the test sequence is of length M, then only time normalizations that map the first frame of the template to the first frame of the test sequence and map the N-th frame of the template sequence to the M-th frame of the test sequence are allowed.
- The warping function (mapping function between the test sequence time to the template sequence time) should increase monotonically. In other words, the sequence of 'events' in both the template and the test sequences should be the same.
- The warping function should be continuous.

Dynamic programming efficiently computes the best warping function and the global warping error. Let us assume that we have chosen to represent the shapes either as a point on a spherical manifold (rotation, scale, translation invariant) or as a subspace that can be identified as a point on the Grassmann manifold (full affine invariance). The manifold nature of the shape-space must be accounted for in the implementation of the DTW algorithm. This implies that during the DTW computation, the local distance measure used must account for the non-Euclidean nature of the shape-space. Therefore, it is only meaningful to use the appropriate distance measure — shape Procrustes in the case of the spherical manifold, and the Grassmann Procrustes in the case of the Grassmann manifold — as described earlier. It is important to note that the Procrustes measure is not a true distance function as it is not symmetric. Moreover, the nature of the constraints makes the DTW algorithm non-symmetric even when we use a symmetric distance for the local feature error. If $A(t)$ and $B(t)$ are two shape sequences, we define the distance between these two sequences $D(A(t), B(t))$ as:

$$D(A(t), B(t)) = DTW(A(t), B(t)) + DTW(B(t), A(t)), \qquad (6.13)$$

where $DTW(A(t), B(t)) = 1/T \sum_{t=1}^{T} d(A(f(t)), B(g(t)))$ (f and g being the optimal warping functions). Such a distance between shape

sequences is symmetric. The isolation property, i.e., $D(A(t), B(t)) = 0$ iff $A(t) = B(t)$, is enforced by penalizing all non-diagonal transitions in the local error metric.

6.5 Applications

Shape matching and shape sequence analyses have several applications in gait-based human recognition, shape retrieval, etc. We next describe a few experiments that demonstrate the strength of these approaches.

6.5.1 Applications in Gait-Based Person Identification

We consider the problem of gait-based person identification, in which our goal is to identify the individual from a video sequence of the individual walking. Given a video sequence, we first perform background subtraction to extract the silhouette of the human. We then uniformly sample points on the outer contour of the silhouette to obtain the pre-shape vector. The procedure for obtaining shapes from the video sequence is illustrated in Figure 6.4. Note that each frame of the video sequence maps to a point on the hyper-spherical shape manifold.

6.5.1.1 Results on the USF database

The USF database [154] consists of 71 people in the gallery. Several covariates, such as camera position, shoe type, surface and time, were varied in a controlled manner to design a set of challenge experiments [154]. The results are evaluated using cumulative match score[1] (CMS) curves and the identification rate. The results for the seven experiments on the USF database are shown in Figure 6.5. These results were originally reported in [217].

6.5.2 Applications to Affine Shape Analysis

We test the Grassmann manifold formulation of affine shape spaces on several standard shape databases for recognition and retrieval tasks.

[1] Plot of percentage of recognition vs rank.

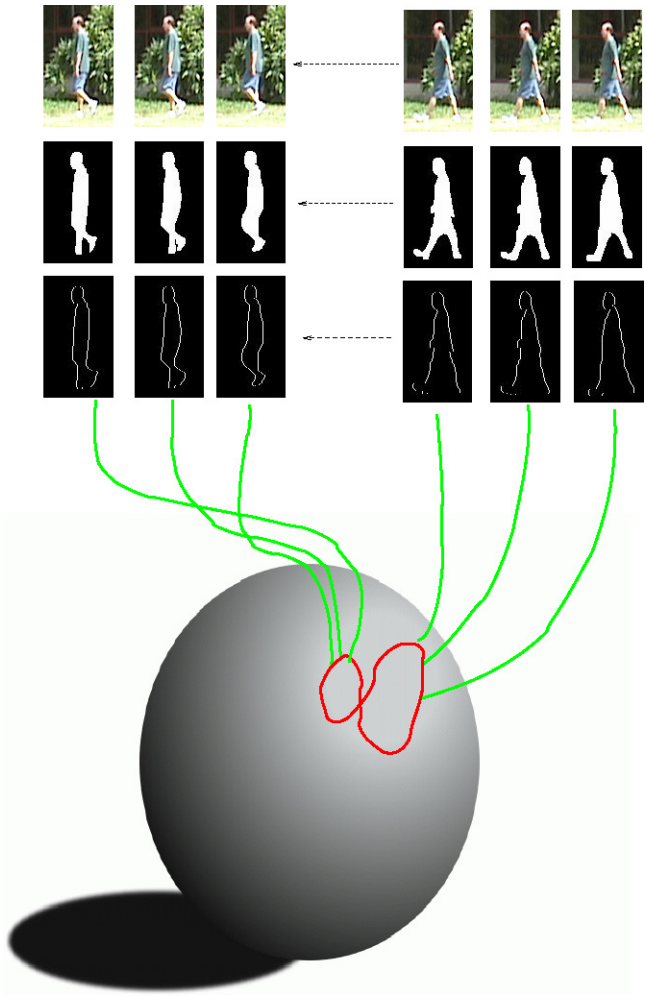


Fig. 6.4 Illustration of the sequence of shapes obtained during a walking cycle. Each frame in the video is processed to obtain a pre-shape vector. These vectors lie on a complex hyper-spherical manifold. Image courtesy [217].

6.5.2.1 Articulation Database

We conducted a retrieval experiment on the articulated shape database from [129]. We used the same test scheme proposed in [129]. The database consists of eight object classes with five examples for each class. For each shape, four top matches are selected and the number of

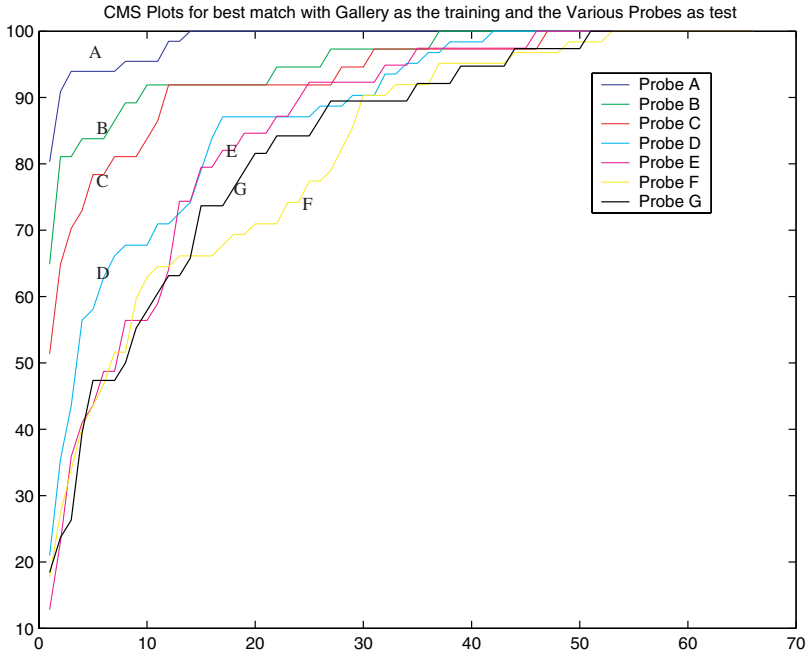


Fig. 6.5 Cumulative Match Scores using Dynamic Time Warping on shape space. These results were first reported in [217].

Table 6.1. Retrieval experiment on articulation data set. Last row is the results obtained using Grassmann manifold Procrustes representation. No articulation invariant descriptors were used. These results were first reported in [207].

Algorithm	Rank 1	Rank 2	Rank 3	Rank 4
SC [129]	20/40	10/40	11/40	5/40
IDSC [129]	40/40	34/40	35/40	27/40
Hashing [23]	40/40	38/40	33/40	20/40
Grassmann Procrustes	38/40	30/40	23/40	17/40

correct hits for ranks $1, 2, 3$, and 4 is reported. Table 6.1 summarizes the results obtained on this data set. The proposed approach compares well with other approaches. It should be noted that this is not a completely fair comparison, as we do not use any articulation-invariant descriptors such as the ones used in [129] and [23]. In spite of this, manifold-based distances perform well.

6.5.2.2 Affine MPEG-7 Database

The strength of the approach lies in affine-invariant representation of shapes, so we conducted a synthetic experiment using the MPEG-7 database. We took one base shape from each of the 70 object classes and created 10 random affine warps of the shapes with varying levels of additive noise. This new set of shapes formed the gallery for the experiment. The generated sample shapes are shown in Figure 6.6. The test set was created by randomly choosing a gallery shape and affine warping it with additive noise. The recognition experiment was performed using the Procrustes distance and the kernel density methods. For comparison, we used the popular shape Procrustes distance [114] as a baseline measure. We also used the 'arc-length' distance of [11], which is the Frobenius norm of the angles between two subspaces. In all cases, the experiments were repeated with 100 Monte Carlo trials for each noise level in order to evaluate the performance robustly. This performance is compared in Figure 6.7, as a function of noise to

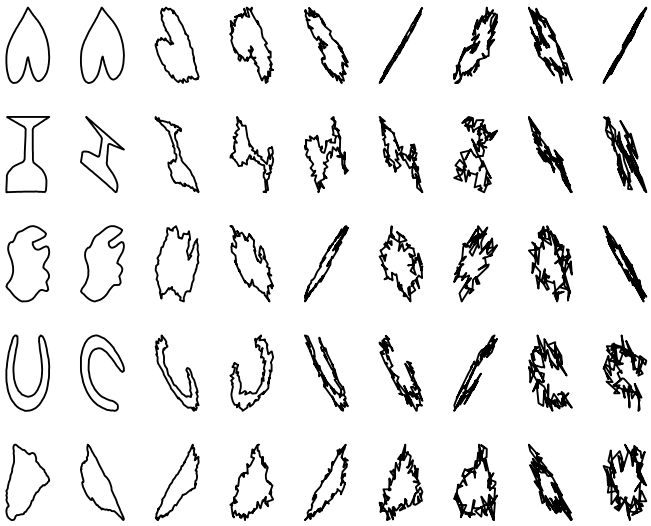


Fig. 6.6 Synthetic data generated from the MPEG database. The first column shows base shapes from the original MPEG data set for five objects. The remaining columns show random affine warps for the base shapes with increasing levels of additive noise. These results were first reported in [207].

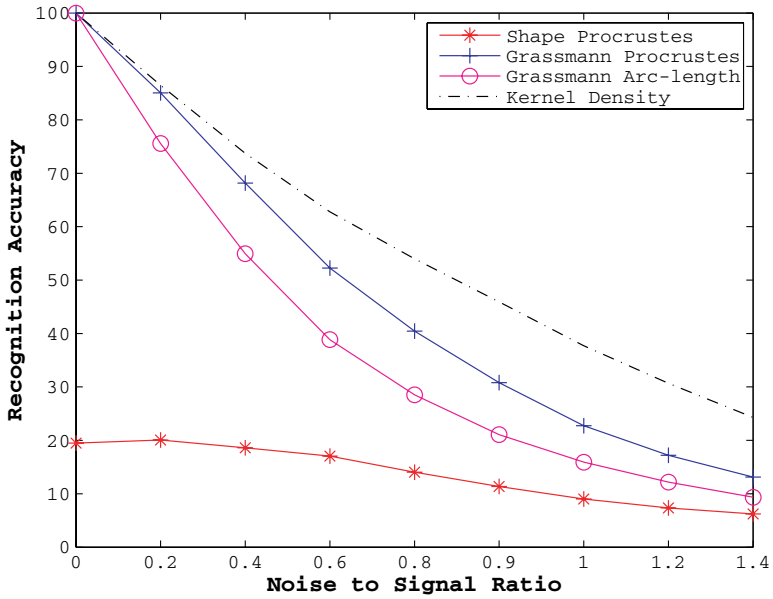


Fig. 6.7 Comparison of recognition performance on MPEG-7 database. For comparison we used the shape Procrustes measure [114] and the Grassmann arc-length distance [11]. Manifold-based methods perform significantly better than direct application of shape Procrustes measure. Among the manifold methods, statistical modeling via kernel methods outperforms the others. These results were first reported in [207].

signal ratio. It can be seen that manifold-based methods perform significantly better than straightforward shape Procrustes measure. Among the manifold methods, the kernel density method outperforms both the Procrustes and the arc-length distance measures. Since the Grassmann manifold-based methods accurately account for the affine variations found in the shape, they outperform simple methods that do not account for affine invariance. Moreover, since the kernel method estimates a pdf for the shapes on the Grassmann manifold, it outperforms distance-based nearest-neighbor classifiers using Grassmann arc-length and Grassmann Procrustes.

6.5.2.3 Sampling from Distributions

Generative capabilities of parametric probability densities can be exploited via appropriate sampling strategies. Once the distribution is

learned, we can synthesize samples from the distribution in a two-step process. We first create a sample from a proposal distribution (we used a matrix-variate normal centered around the class mean), then we use an accept–reject strategy to generate the final shape [41]. We show a sampling experiment using this technique. For this experiment, we chose one shape from each of the object classes in the MPEG-7 database and corrupted it with additive noise to generate several noisy samples for each class. We used the Grassmann representation of points as idempotent projection matrices. Then, we learned a parametric Langevin distribution [41] on the Grassmann manifold for each class. Note that the distribution is learned on the Grassmann manifold, hence, a sample from the distribution represents a subspace in the form of a projection matrix. To generate an actual shape, we need to first choose a two-frame for the generated subspace which can be performed via singular value decomposition (SVD) of the projection matrix. Once the two-frame is chosen, actual shapes can be generated by choosing random coordinates in the two-frame. We show sampling results in Figure 6.8.

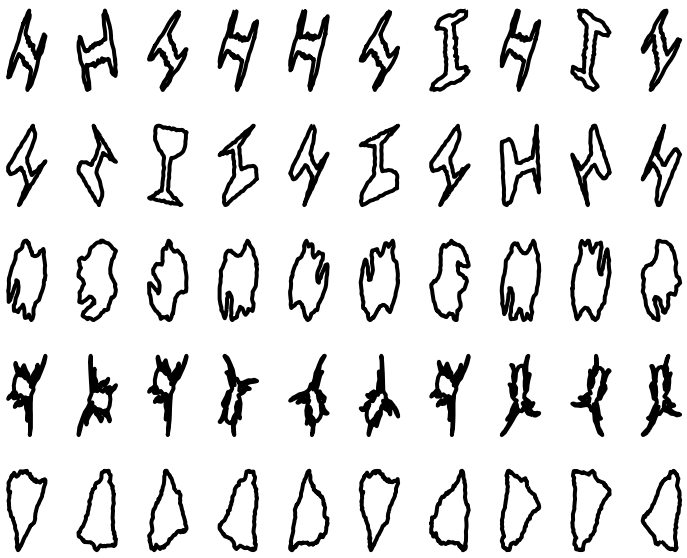


Fig. 6.8 Samples generated from estimated class conditional densities for a few classes of the MPEG data set. These results were first reported in [207].

6.5.3 Applications to Spatio-Temporal Dynamical Modeling

The Grassmann manifold models described above provide a rich class of statistical methods with applications extending beyond shape classification. We discuss several other applications in this section that involve comparison of points on the Grassmann manifold. Specifically, we discuss a Grassmann manifold formulation of autoregressive moving average (ARMA) models and show how these methods can be applied to activity analysis and video-based face recognition. A wide variety of time series data such as dynamic textures, human joint angle trajectories, shape sequences, and video-based face recognition etc are frequently modeled as autoregressive and moving average (ARMA) models [3, 22, 217, 188]. The ARMA model equations are given by:

$$f(t) = Cz(t) + w(t), \quad w(t) \sim N(0, R), \tag{6.14}$$

$$z(t+1) = Az(t) + v(t), \quad v(t) \sim N(0, Q), \tag{6.15}$$

where z is the hidden state vector, A the transition matrix, and C the measurement matrix. f represents the observed features while w and v are noise components modeled as normal with 0 mean and covariances R and Q, respectively. Closed form solutions for learning the model parameters (A, C) from the feature sequence $(f_{1:T})$ are widely used in the computer vision community [188]. Let observations $f(1), f(2), \ldots, f(\tau)$ represent the features for the time indices $1, 2, \ldots, \tau$. Let $[f(1), f(2), \ldots, f(\tau)] = U\Sigma V^T$ be the singular value decomposition of the data. Then $\hat{C} = U, \hat{A} = \Sigma V^T D_1 V (V^T D_2 V)^{-1} \Sigma^{-1}$, where $D_1 = [0\ 0; I_{\tau-1}\ 0]$ and $D_2 = [I_{\tau-1}\ 0; 0\ 0]$.

For comparison of two models, we compare the subspaces spanned by the respective observability matrices [43]. The extended observability matrix for a model (A, C) is given by:

$$O_\infty^T = \left[C^T, (CA)^T, (CA^2)^T, \ldots, (CA^{n-1})^T \ldots\right]. \tag{6.16}$$

Thus, a linear dynamical system can be alternately identified as a point on the Grassmann manifold, corresponding to the column space of its observability matrix. In the experiments shown here (and originally reported in [207]), the extended observability matrix was approximated

by the finite observability matrix as is commonly done [175]:

$$O_n^T = \left[C^T, (CA)^T, (CA^2)^T, \ldots (CA^{n-1})^T\right]. \tag{6.17}$$

As already discussed in Section 6.3.3.2, comparison of two points on the Grassmann manifold can be performed by using the Procrustes metric (denoted as NN-Procrust). Moreover, if several observation sequences are available for each class, then one can learn the class conditional distributions on the Grassmann manifold using kernel density methods. Maximum likelihood classification can be performed for each test instance using these class conditional distributions (denoted as Kernel-Procrust).

6.5.3.1 Activity Recognition

We performed a recognition experiment on the publicly available INRIA data set [221]. The data set consists of 10 actors performing 11 actions, each action executed three times at varying rates while freely changing orientation. We used the view-invariant features as proposed in [221]. Specifically, we used the $16 \times 16 \times 16$ circular FFT features proposed by [221]. Each activity was modeled as a linear dynamical system. Testing was performed using a round-robin experiment, in which activity models were learned using nine actors and tested on the remaining actor. For the kernel method, all available training instances per class were used to learn a class conditional kernel density, as described in Section 6.3.3.3. In Table 6.2, we show the recognition results obtained using four methods. The first column shows the results obtained using dimensionality reduction approaches of [221] on $16 \times 16 \times 16$ features. Weinland et al. [221] report recognition results using a variety of dimensionality reduction techniques (PCA, LDA, Mahalanobis), and we choose the best performance from their experiments (denoted 'Best Dim. Red.') which were obtained using $64 \times 64 \times 64$ circular FFT features. The third column corresponds to the method of using subspace angles-based distance between dynamical models [43]. Column 4 shows the nearest-neighbor classifier performance using Procrustes distance measure on the Stiefel manifold ($16 \times 16 \times 16$ features). We see that

Table 6.2. Comparison of view-invariant recognition of activities in the INRIA data set using (a) Dimensionality reduction approaches of [221] on $16 \times 16 \times 16$ features, (b) Best performance obtained in [221] on $64 \times 64 \times 64$ features, (c) Nearest neighbor using Procrustes distance ($16 \times 16 \times 16$ features), (d) Maximum likelihood using kernel density methods ($16 \times 16 \times 16$ features). These results were first reported in [207].

Activity	Dim. Red. [221] 16^3 volume	Best Dim. Red. [221] 64^3 volume	Subspace Angles 16^3 volume	NN-Procrust 16^3 volume	Kernel-Procrust 16^3 volume
Check Watch	76.67	86.66	93.33	90	100
Cross Arms	100	100	100	96.67	100
Scratch Head	80	93.33	76.67	90	96.67
Sit Down	96.67	93.33	93.33	93.33	93.33
Get Up	93.33	93.33	86.67	80	96.67
Turn Around	96.67	96.67	100	100	100
Walk	100	100	100	100	100
Wave Hand	73.33	80	93.33	90	100
Punch	83.33	96.66	93.33	83.33	100
Kick	90	96.66	100	100	100
Pick Up	86.67	90	96.67	96.67	100
Average	88.78	93.33	93.93	92.72	98.78

Table 6.3. Comparison of video-based face recognition approaches (a) ARMA system distance, (b) Grassmann Procrustes distance, and (c) Kernel density on Grassmann manifold. These results were first reported in [207]

	Test condition	ARMA model distance [43]	Procrustes	Kernel density
1	Gallery1, Probe2	81.25	93.75	93.75
2	Gallery2, Probe1	68.75	81.25	93.75
3	Average	75%	87.5%	93.75%

the manifold Procrustes distance performs as well as ARMA model distance [43]. However, statistical modeling of class conditional densities for each activity using kernel density methods on the Grassmann manifold leads to a significant improvement in recognition performance. Note that even though the manifold approaches use only $16 \times 16 \times 16$ features they outperform other approaches that use higher resolution ($64 \times 64 \times 64$ features), as shown in Table 6.2.

6.5.3.2 Video-Based Face Recognition

Video-based face recognition (FR) by modeling the 'cropped video' either as dynamical models [3] or as a collection of PCA subspaces

Fig. 6.9 Sample images from the skating video from [220].

Fig. 6.10 Shown above are a few sequences from Cluster 1. Each row shows contiguous frames of a sequence. We see that this cluster dominantly corresponds to 'Sitting Spins'. Image best viewed in color. These results were first reported in [209].

[125] has recently gained popularity because of its ability to recognize faces from low-resolution videos. In this case, we use only the C matrix of the ARMA model or PCA subspace as the distinguishing model parameter, as the C matrix encodes the appearance of the face. The

C matrices are orthonormal, hence are points on the Stiefel manifold. For recognition applications, the important information is encoded in the subspace spanned by the C matrix. Hence, we identify the model parameters (Cs) as points on the Grassmann manifold, and both Procrustes distance and kernel density methods are directly applicable. We tested our method on the data set used in [3]. The data set consists of face videos for 16 subjects with two sequences per subject. Subjects arbitrarily change head orientation and expressions. The illumination conditions differed widely for the two sequences of each subject. For each subject, one sequence was used as the gallery and the other formed the probe. The experiment was repeated by swapping the gallery and the probe data. The recognition results are reported in Table 6.3. For kernel density estimation, the available gallery sequence for each actor was split into three distinct sequences. As seen in the last column, the kernel-based method outperforms the other approaches.

Fig. 6.11 Shown above are a few sequences from Cluster 2. Each row shows contiguous frames of a sequence. Notice that this cluster dominantly corresponds to 'Standing Spins'. Image best viewed in color. These results were first reported in [209].

6.5.3.3 Video Clustering

In [209], the ARMA model was used to perform activity based clustering of video sequences. The algorithm clusters a long video sequence into several clusters by modeling short segments of the video as outputs of an ARMA model. Further, by combining principles of geometric invariance with system identification, limited view and execution-rate invariance were also built into the system. Here, we describe a clustering experiment on the figure skating data set from [220]. These data are very challenging since these are unconstrained and involves rapid motion of the skater and real-world motion of the camera including pan, tilt, and zoom. Some representative frames from the raw video are shown in Figure 6.9. To obtain low-level features, color models of the foreground and background were built using normalized color histograms. The color histograms are used to segment the background and foreground pixels. Median filtering followed by connected component

Fig. 6.12 Shown above are a few sequences from Cluster 4. Each row shows contiguous frames of a sequence. This cluster dominantly corresponds to 'Camel Spins'. Image best viewed in color. These results were first reported in [209].

analysis is performed to reject small isolated blobs. From the segmented results, a bounding box enclosing the foreground pixels is estimated. Temporal smoothing of the bounding box parameters is performed to remove jitter. The final feature is a rescaled binary image of the pixels inside the bounding box. We show some sample sequences in the obtained clusters in Figures 6.10–6.12. We observe that Clusters 1–3 correspond dominantly to 'Sitting Spins', 'Standing Spins', and 'Camel Spins' respectively. More detailed results can be seen in [209].

7

Future Trends

We have examined several interesting computer vision applications such as target tracking, structure from motion, shape recovery, face recognition, gait-based person identification, and video-based activity recognition. We explored the fundamental connections between these various problems in terms of the geometric modeling assumptions necessary to solve them and studied the statistical techniques that enable robust solutions to these problems. Recent research in statistical methods have made significant progress, and, of the various statistical methods that have recently emerged, we highlight a few recent trends that we believe will have significant impact on various computer vision tasks.

7.1 New Data Processing Techniques: Non-linear Dimensionality Reduction

Images and video data lead to extremely high-dimensional data sets and in order to represent, visualize, analyze, interpret, and process such high-dimensional data, one needs to encapsulate the salient characteristics of the data in a lower dimensional representation. Traditionally, this has been performed using linear dimensionality reduction techniques such as principal component analysis (PCA) [185]

or independent component analysis (ICA) [96]. Such linear dimensionality reduction methods are limited in applicability for vision applications, because the geometric constraints imposed by the imaging device and the lighting characteristics lead to non-linear constraints on image and video data. Recent research in the fields of manifold learning and non-linear dimensionality reduction (NLDR) has led to an explosion of new results and algorithms designed to tackle such non-linear embeddings of the high-dimensional data.

NLDR approaches such as the locally linear embedding [171] (LLE), Hessian LLE [53], Isomap [201], the Laplacian eigenmap [13], and local tangent space alignment (LTSA) [230] construct and represent the high-dimensional data using a low-dimensional representation. The parameters of the representation are obtained by optimizing an appropriate cost function that leads to embeddings that are 'Euclidean-like' locally, but globally are highly nonlinear. Moreover, most of these techniques also have an elegant graph formulation that explicitly describes the local properties that these approaches preserve. Such approaches for NLDR find several natural and compelling applications in vision tasks, as both image and video data are inherently high-dimensional and lend themselves to non-linear dimensionality reduction.

7.1.1 Applications in Classification

The success of dimensionality reduction approaches in vision tasks comes far since the early days of eigenfaces [210]. One of the drawbacks of early classifiers was that the complexity of the decision boundaries increased with increase in the dimensionality of data. Therefore, the number of training samples required to learn these decision boundaries was large. One way to address this problem is via dimensionality reduction, where the high-dimensional data points embedded into a lower dimensional space and the classifiers are trained on this lower dimensional space. If the dimensionality reduction is done 'properly' then one could obtain reduction in the complexity of the classifier without sacrificing performance. Non-linear dimensionality reduction techniques have found applications in face recognition [5, 171, 228] and in handwriting recognition [120].

7.1.2 Applications in Video Processing

Video sequences are also naturally amenable to analysis using nonlinear dimensionality reduction approaches. As an example consider action analysis and activity recognition. The goal is to recognize the pose of the subject from images, then use the knowledge about pose to perform activity recognition. Since the human body can be modeled as a kinematic chain with small degrees of freedom and, in most cases, the camera is assumed static, this means that the obtained silhouettes of the subject span a low-dimensional space. If NLDR approaches were used to extract this low-dimensional representation, then problems such as pose recognition and activity recognition could be solved using these projections, as opposed to the original imaging data. Similar ideas and approaches for pose recognition have been gaining popularity in recent years [40, 62, 63].

Another interesting application of manifold learning approaches in video processing comes from visualization. Multiple cameras are used constantly in security and surveillance applications, and a growing need exists to visualize the data in a form that emphasizes the content of the data rather than presentation in a time-ordered fashion. NLDR approaches are used to cluster the data in a low-dimensional space, and the clustered results are then presented to the human operators. In Figure 7.1 we show that an individual's actions cluster into trajectories when embedded in a low-dimensional Laplacian eigenspace.

NLDR approaches may also apply in extracting the inherent underlying structure of 3D data obtained using multiple cameras. In [199], it is shown that the Laplacian eigenspace maps voxels on long articulated chains to smooth 1D curves. This facilitates discrimination of the limbs and helps to segment the voxel data into limbs, head, and trunk. Figure 7.2 illustrates this idea for segmenting and registering 3D voxel data of a human.

7.2 New Hardware and Cameras: Compressive Sensing

Recent advances in sparse approximations and reconstruction algorithms that use sparsity priors have huge relevance to imaging and vision applications. Images and video are typically high-dimensional

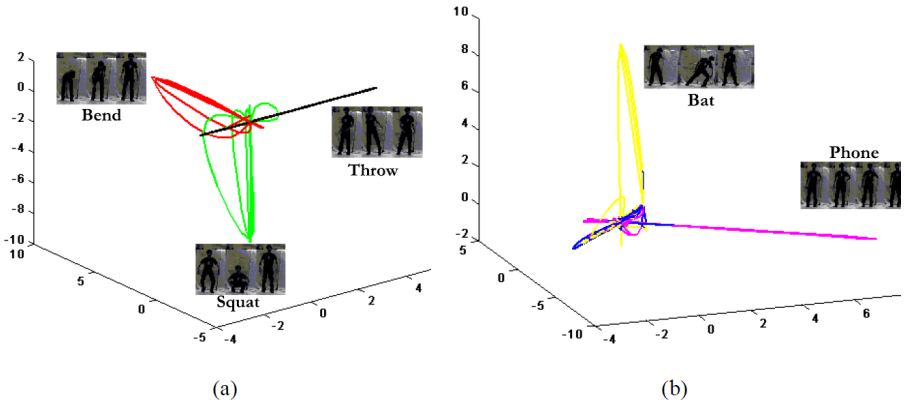


Fig. 7.1 Example of NLDR used for visualizing the actions performed in a video sequence (courtesy: [208]). (a) Visualization of the clusters in Laplacian space dimensions 1–3. (b) Visualization of clusters in Laplacian space dimensions 4–6. Best viewed in color.

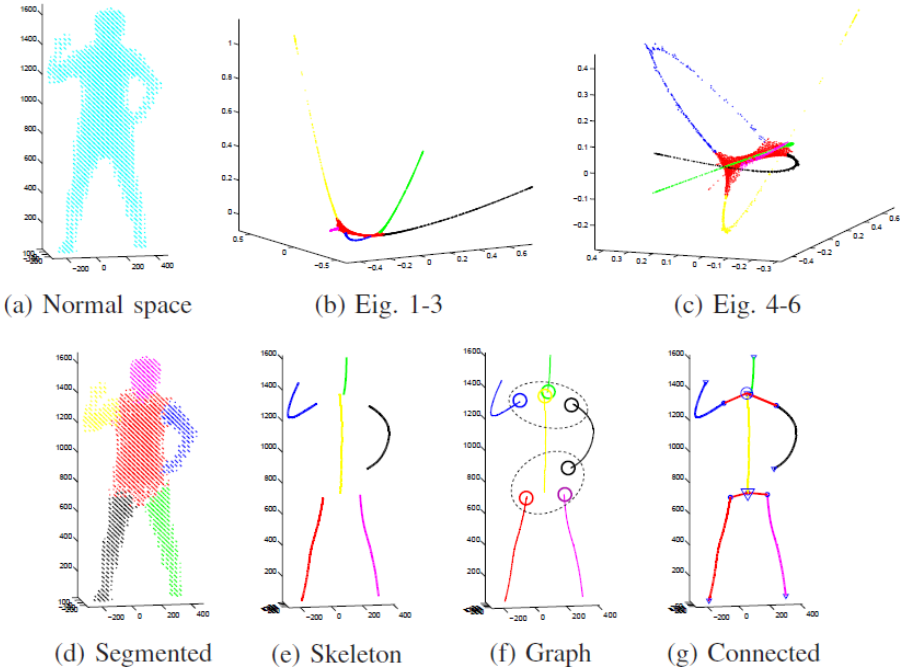


Fig. 7.2 Segmentation and registration in 3D using Laplacian eigenmaps (courtesy: [199]). The nodes are segmented in LE (b–c). The labels are represented in the original 3D space in (d). The computed skeleton is presented in (e) and the two joints in (f). The correct registration is shown in (g).

data, but simple and sometimes even linear, transformations of images and video lead to sparse coefficients. Under this assumption, recent advances in the field of compressive sensing [10, 33, 34, 52] can lead to interesting consequences for vision algorithms.

7.2.1 Compressive Sensing Theory

Estimating a K-sparse vector s that satisfies $y = As + e$ (where e is the noise), can be formulated as the following L_0 optimization problem:

$$(P0): \qquad \min ||s_0|| \quad s.t. \quad ||y - As||_2 \leq \epsilon, \tag{7.1}$$

where the L_0 norm counts the number of non-zero elements. This is typically an NP-hard problem. However, the surprising equivalence between L_0 and L_1 norm for such systems [34] allows us to reformulate the problem as one of L_1 norm minimization:

$$(P1): \qquad \min ||s_1|| \quad s.t \quad ||y - As||_2 \leq \epsilon \tag{7.2}$$

with ϵ being a bound for the measurement noise e. It has been shown that the solution to $(P1)$ is, with high probability, the K-sparse solution that we seek. Further, $(P1)$ is a convex program for which there are efficient polynomial-time solutions, as well as effective numerical implementations.

A wide variety of reconstruction algorithms can solve the convex optimization problem $(P1)$. One condition must be satisfied so that the reconstruction algorithm is robust and accurate. The effective mixing matrix A should have the restricted isometry property (RIP) of order 3K or more, so a K-sparse signal can be effectively reconstructed.

7.2.2 Application to Image Capture

Since it can be shown that images are sparse in the wavelet basis, we can reconstruct their wavelet coefficients if we can measure arbitrary linear combinations of the images using an optical device. The Rice compressive sensing camera [200] achieves this by using a digital micromirror array device (DMD). The DMD array can essentially turn each pixel 'ON' or 'OFF', and a single photosensor inside the camera measures the sum of all the light intensity that enters the camera. So,

each measurement of the photosensor is one projection of the image. By programming the DMD device, we can measure multiple independent projections of the image. Compressive sensing techniques can then be used for reconstructing the image from the linear measurements.

7.2.3 Application in Reflectance Field Measurement

Acquiring the reflectance field of real objects is an important problem often encountered in solving several imaging, vision, and graphics tasks such as relighting. Typical methods for capturing such reflectance fields are based on acquiring images with a single light source direction turned on. Recently, Schechner et al. [181] have shown that using multiple light sources per each acquired image and performing a linear inversion to recover the reflectance field results in higher signal to noise ratios of the captured reflectance fields. Nevertheless, the number of images to be acquired to infer the reflectance field remains identical to the number of illumination sources. However, reflectance fields are inherently sparse in various bases. We can show that depending on the surface characteristic (Lambert, Phong) the reflectance field of a surface is highly compressible in the DCT or the Haar wavelet basis. Using this as a motivation, similar to the single pixel camera (SPC), we show that the reflectance field associated with a scene can be obtained by just a few random projections. Figure 7.3 shows reconstructed reflectance fields obtained from such an approach.

7.2.4 Applications in Video

In security and surveillance applications, sometimes the goal is not to capture the entire video but just the moving objects in the scene. This process, also called background subtraction, is usually done by first capturing the entire video data, then performing statistical analysis to determine what pixels in the image are moving. The number of moving pixels in an image is usually much smaller than the total number of pixels in an image. This means that such background subtracted images are inherently sparse in the image domain. In [36], it is shown to be possible to reconstruct background silhouettes directly from compressive measurements of images. Further, it is possible to obtain background

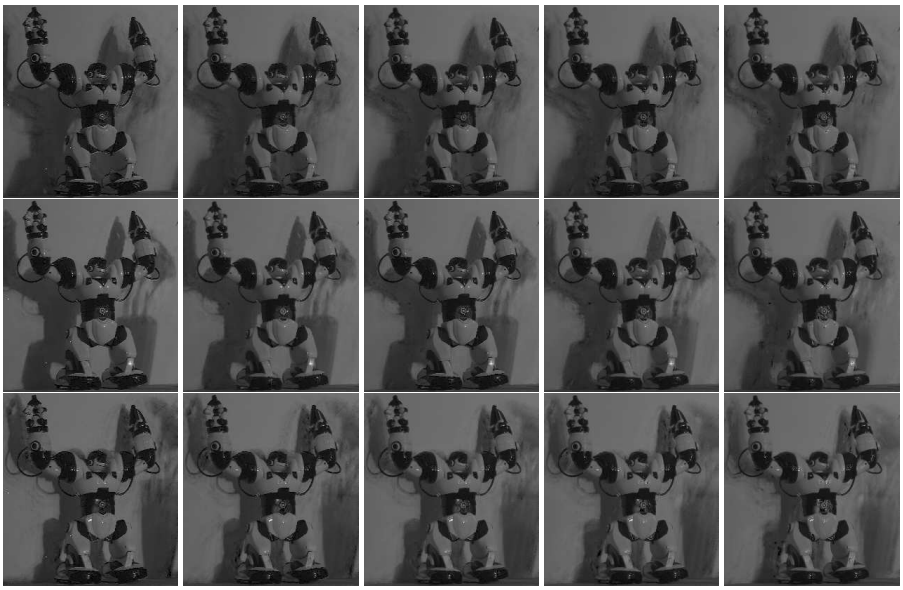


Fig. 7.3 Results of reflectance field acquisition obtained from compressed measurements of various compression factors. Each column corresponds to a different compression factor (L to R: 1.3, 1.7, 2, 2.7, 4), and each row corresponds to a different source direction. Images courtesy [176].

subtraction silhouettes at compression ratios far higher than required to reconstruct the original images. Figure 7.4 illustrates this idea in detail.

Similarly, Veeraraghavan et al. [216] shows that one can use a simple temporal modulation in the aperture to convert an off-the-shelf video camera into a high-speed camera. This is done by modeling the high speed video observed at a pixel as being sparse in the Fourier domain. This is possible since the signal being observed is periodic, hence, in the ideal case, its Fourier transform consists of a train of impulses.

7.3 New Mathematical Tools: Analytic Manifolds

As we have seen so far, applications in computer vision involve the study of geometric scenes and their interplay with physical phenomena such as illumination and motion. When these scenes are imaged using cameras, the observed appearances obey certain mathematical

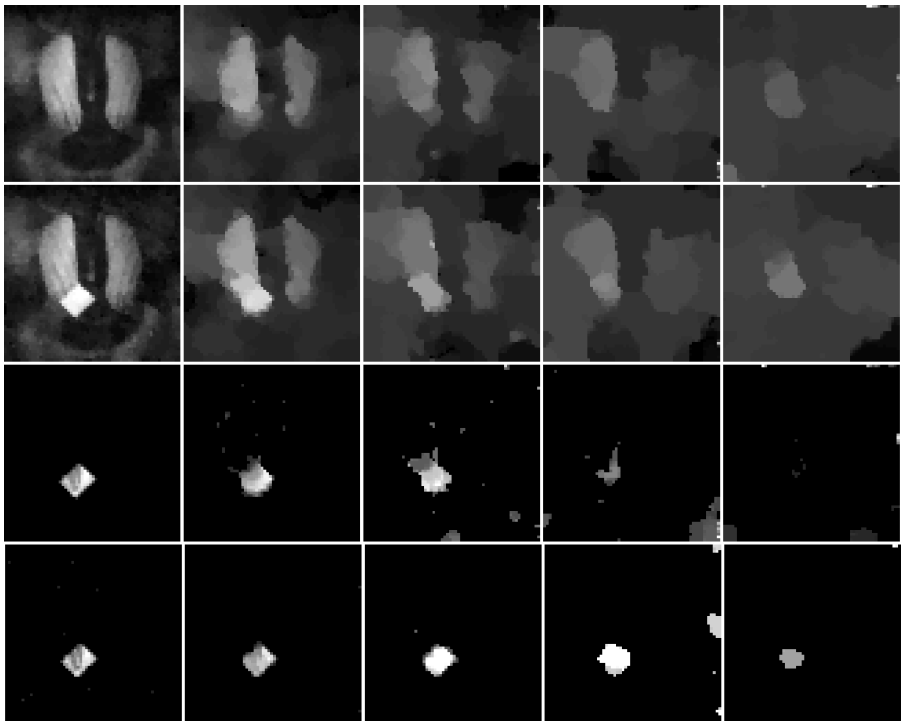


Fig. 7.4 Background subtraction experimental results using an SPC (figure courtesy [36]). Reconstruction of background image (top row) and test image (second row) from compressive measurements. Third row: conventional subtraction using the above images. Fourth row: reconstruction of difference image directly from compressive measurements. The columns correspond to measurement rates M/N of 50%, 5%, 2%, 1%, and 0.5%, from left to right. Background subtraction from compressive measurements is feasible at lower measurement rates than standard background subtraction.

constraints that are induced by the underlying physical constraints. Examples include the observation that images of a convex object under all possible illumination conditions lie on the so-called 'illumination-cone' [77]. Images taken under a stereo pair are constrained by the epipolar geometry of the cameras [89]. Similarly, the 3D pose of the human head is parameterized by three angles — hence, under constant illumination and expression, the observed face of a human under different viewing directions lies on a 3D manifold. In a particular application, if the physical and mathematical constraints are well-understood, such as in epipolar geometry and illumination modeling, then one can

design accurate modeling and inference techniques derived from this understanding.

In several applications such as gait-based human ID, activity recognition, shape-based dynamics modeling, and video-based face recognition, some of the constraints that arise have a special form. These special constraints can often be expressed in the form of an equation with some smoothness criteria. Such constraints can be formally defined as manifolds. Rather informally, we will assume that a 'manifold' is defined as a set of points in $\mathbb{R}^n$ that satisfy an equation $f(x) = 0$ (with appropriate conditions on $f()$). For example, the set of points that satisfy the equation $f(x) = x^T x - 1 = 0$ is the unit hyper-spherical manifold in $\mathbb{R}^n$. We provide specific examples of various analytical manifolds found in different applications of computer vision below.

1. **Feature Manifolds:** Image and video understanding typically begins with the extraction of some specific features from raw data. Examples of these features include background-subtracted images, shapes, intensity features, motion vectors, etc. These features extracted from the videos might satisfy certain geometric and photometric constraints. The feature space deals with understanding and characterizing the geometry of features that can be extracted from videos. The study of this space then enables appropriate modeling methodologies to be designed. Consider the example of the shape feature. Shapes in images are commonly described by a set of landmarks on the object being imaged. After appropriate translation, scale and rotation normalization it can be shown that shapes reside on a complex spherical manifold. Further, by factoring out all possible affine transformations, it can be shown that shapes reside on a Grassmann manifold. More recently, shapes have been considered to be continuous closed planar curves. The space of such curves can also be characterized as a manifold [193]. A recently developed formulation of using the covariance of features in image-patches has found several applications such as texture classification [212], pedestrian detection [213],

and tracking [157]. The Riemannian geometry of covariance matrices was exploited effectively in all these applications to design state-of-the-art algorithms.

2. **Model Spaces:** After features are extracted from imagery, the next step is to describe the features using appropriate spatio-temporal models. One specific example of this is modeling a feature sequence as realizations of dynamical systems. Examples include dynamic textures, human joint angle trajectories, and silhouette sequences [188]. One popular dynamical model for such time-series data is ARMA model. The space spanned by the columns of the observability matrix of the ARMA model can be identified as a point on the Grassmann manifold. Time-varying and switching linear dynamical systems can then be interpreted as paths on the Grassmann manifold [206, 207]. The traditionally used linear subspace models of data can be studied by understanding their structure as a Grassmann manifold. This enables application-driven dimensionality reduction approaches where the cost function to be optimized can be quite arbitrary [194], such as enforcing sparsity in coefficients, or improving discrimination in classification.
3. **Transformation Spaces:** Finally, the transformation space studies all the possible manifestations of the same semantic content, and is thus important to achieve invariance to factors such as view changes and execution-rate changes. The space of execution-rate variations in human activities, which is modeled as temporal warps of feature trajectories, is the space of positive and monotonically increasing functions mapping the unit-interval to the unit-interval. The derivatives of warping functions can be interpreted as probability density functions. The square-root form of pdfs can further be described as a sphere in the space of functions. This formalism is used to achieve execution-rate invariance in human activity modeling by Veeraraghavan et al. [218]. Variability in sampling closed planar curves gives rise to variations in observed feature points on shapes [142]. This variability can

also be modeled as a sphere in the space of functions (also known as a Hilbert sphere).

As these examples illustrate, manifolds arise quite naturally in several vision-based applications. The area of statistics and inference on manifolds has seen a large growth in recent years. Many of the ideas have been formally introduced and advanced through the efforts of many researchers. In several applications, the superior performance of algorithms that exploit the geometric properties of the underlying manifold has been demonstrated.

Acknowledgments

The monograph has benefited immensely from the valuable comments of the reviewers and the editor. We are particularly indebted to Prof. Bob Gray and Reviewer A for the thorough proofreading, and the painstakingly detailed review.

We also thank former students and collaborators — Ted Broida (deceased), Gem-Sun Young, Sridhar Srinivasan, Gang Qian, Amit K. Roy-Chowdhury, Kevin Zhou, Jie Shao, Jian Li, Gaurav Aggarwal — for letting us draw upon their work, thus making this monograph possible.

We would further like to thank Prof. Anuj Srivastava (Dept. of Statistics, Florida State University, Tallahassee) and Prof. Ankur Srivastava (Electrical and Computer Engineering, University of Maryland, College Park) for their helpful discussions.

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre, "Riemannian geometry of Grassmann manifolds with a view on algorithmic computation," *Acta Applicandae Mathematicae: An International Survey Journal on Applying Mathematics and Mathematical Applications*, vol. 80, no. 2, pp. 199–220, 2004.

[2] G. Adiv, "Determining three-dimensional motion and structure from optical flow generated by several moving objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, no. 4, pp. 384–401, 1985.

[3] G. Aggarwal, A. Roy-Chowdhury, and R. Chellappa, "A system identification approach for video-based face recognition," in *Proceedings of the 17th International Conference on Pattern Recognition*, vol 4, pp. 175–178, 2004.

[4] G. Aggarwal, A. Veeraraghavan, and R. Chellappa, "3D facial pose tracking in uncalibrated videos," *Lecture Notes in Computer Science*, vol. 3776, p. 515, 2005.

[5] O. Arandjelovic and R. Cipolla, "Face recognition from face motion manifolds using robust kernel resistor-average distance," in *Proceedings of IEEE Workshop on Face Processing in Video*, pp. 88–88, 2004.

[6] E. M. Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem, and J. S. B. Mitchell, "An efficiently computable metric for comparing polygonal shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 3, pp. 209–216, 1991.

[7] A. Azarbayejani and A. Pentland, "Recursive estimation of motion structure and focal length," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 6, pp. 562–575, 1995.

[8] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," in *Proceedings of IEEE International Conference on Computer Vision*, pp. 1–8, 2007.

[9] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association*. San Diego, CA, USA: Academic Press Professional, Inc, 1987.

[10] R. G. Baraniuk, "Compressive sensing," *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 118–121, 2007.

[11] E. Begelfor and M. Werman, "Affine invariance revisited," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 2087–2094, 2006.

[12] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projections," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[13] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[14] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509–522, 2002.

[15] R. Berthilsson, "A statistical theory of shape," *Lecture Notes in Computer Science*, vol. 1451, pp. 677–686, 1998.

[16] J. Besag, "Statistical analysis of non-lattice data," *The Statistician*, vol. 24, no. 3, pp. 179–195, 1975.

[17] J. Besag, "On the statistical analysis of dirty pictures," *Journal of the Royal Statistical Society B*, vol. 48, pp. 259–302, 1986.

[18] J. Besag, "On the statistical analysis of dirty pictures," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 48, no. 3, pp. 259–302, 1986.

[19] J. Besag, "Markov chain Monte Carlo for statistical inference," Technical Report, Seattle: Center for Statistics and Social Sciences, University of Washington, 2001.

[20] G. Bilbro, R. Mann, T. K. Miller, W. E. Snyder, D. E. Van den Bout, and M. White, "Optimization by mean field annealing," in *Proceedings of Advances in Neural Information Processing Systems*, vol. 1, pp. 91–98, 1989.

[21] C. M. Bishop, *Pattern Recognition and Machine Learning*. NJ, USA: Springer-Verlag New York, 2006.

[22] A. Bissacco, A. Chiuso, Y. Ma, and S. Soatto, "Recognition of human gaits," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 52–57, 2001.

[23] S. Biswas, G. Aggarwal, and R. Chellappa, "Efficient indexing for articulation invariant shape matching and retrieval," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007.

[24] J. Black, T. Ellis, and P. Rosin, "Multi view image surveillance and tracking," in *Proceedings of Workshop on Motion and Video Computing*, pp. 169–174, 2002.

[25] V. Blanz and T. Vetter, "Face recognition based on fitting a 3D morphable model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1063–1074, 2003.

[26] H. Blum and R. Nagel, "Shape description using weighted symmetric axis features," *Pattern Recognition*, vol. 10, no. 3, pp. 167–180, 1978.

[27] R. M. Bolle and D. B. Cooper, "Bayesian recognition of local 3-D shape by approximating image intensity functions with quadric polynomials," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 4, pp. 418–429, 1984.

[28] R. M. Bolle and D. B. Cooper, "On optimally combining pieces of information, with application to estimating 3-D complex-object position from range data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 5, pp. 619–638, 1986.

[29] F. L. Bookstein, "Size and shape spaces for landmark data in two dimensions," *Statistical Science*, vol. 1, no. 2, pp. 181–222, 1986.

[30] T. Broida and R. Chellappa, "Performance bounds for estimating three-dimensional motion parameters from a sequence of noisy images," *Journal of Optical Society of America A*, vol. 6, no. 6, pp. 879–889, 1989.

[31] T. J. Broida, S. Chandrashekhar, and R. Chellappa, "Recursive 3-D motion estimation from a monocular image sequence," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, no. 4, pp. 639–656, 1990.

[32] M. J. Brooks, *Shape from Shading (Ed. B. K. P. Horn).* Cambridge, MA, USA: MIT Press, 1989.

[33] E. Candes, "Compressive sampling," in *Proceedings of the International Congress of Mathematicians*, vol. 3, pp. 1433–1452, 2006.

[34] E. Candes and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse Problems*, vol. 23, no. 3, pp. 969–985, 2007.

[35] G. Casella, R. L. Berger, and R. L. Berger, *Statistical Inference.* CA: Duxbury Pacific Grove, 2002.

[36] V. Cevher, A. Sankaranarayanan, M. F. Duarte, D. Reddy, R. G. Baraniuk, and R. Chellappa, "Compressive sensing for background subtraction," in *Proceedings of European Conference on Computer Vision*, pp. 12–18, 2008.

[37] R. Chellappa, "Two-dimensional discrete Gaussian Markov random field models for image processing," in *Progress in Pattern Recognition 2*, (L. N. Kanal and A. Rosenfeld, eds.), pp. 79–112, New York: Elsevier, 1985.

[38] R. Chellappa and R. Kashyap, "Texture synthesis using 2-D noncausal autoregressive models," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 33, no. 1, pp. 194–203, 1985.

[39] C. C. Chen, "Improved moment invariants for shape discrimination," *Pattern Recognition*, vol. 26, no. 5, pp. 683–686, 1993.

[40] D. Chen, J. Zhang, S. Tang, and J. Wang, "Freeway traffic stream modeling based on principal curves and its analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 246–258, 2004.

[41] Y. Chikuse, *Statistics on Special Manifolds.* Springer Verlag, 2003.

[42] A. K. R. Chowdhury and R. Chellappa, "Face reconstruction from monocular video using uncertainty analysis and a generic model," *Computer Vision and Image Understanding*, vol. 91, no. 1–2, pp. 188–213, 2003.

[43] K. D. Cock and B. De Moor, "Subspace angles and distances between ARMA models," in *Proceedings of the International Symposium of Mathematical Theory of Networks and Systems*, vol. 1, 2000.

[44] F. S. Cohen and D. B. Cooper, "Simple parallel hierarchical and relaxation algorithms for segmenting noncausal Markovian random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 2, pp. 195–219, 1987.

[45] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 142–149, 2000.

[46] A. Criminisi, *Accurate Visual Metrology from Single and Multiple Uncalibrated Images*. Springer Verlag, 2001.

[47] A. Criminisi, I. Reid, and A. Zisserman, "Single view metrology," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 123–148, 2000.

[48] G. R. Cross and A. K. Jain, "Markov random field texture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, no. 1, pp. 25–39, 1983.

[49] M. H. DeGroot, *Optimal Statistical Decisions*. Wiley-Interscience, 2004.

[50] Q. Delamarre and O. Faugeras, "3D articulated models and multi-view tracking with silhouettes," in *Proceedings of IEEE International Conference on Computer Vision*, vol. 2, pp. 716–721, 1999.

[51] H. Derin and H. Elliott, "Modeling and segmentation of noisy and textured images using Gibbs random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 1, pp. 39–55, 1987.

[52] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[53] D. L. Donoho and C. Grimes, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data," in *Proceedings of the National Academy of Sciences*, pp. 5591–5596, 2003.

[54] A. Doucet, "On sequential simulation-based methods for Bayesian filtering," Technical Report, Department of Engineering, University of Cambridge, 1998.

[55] A. Doucet, N. D. Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag, 2001.

[56] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.

[57] I. Dryden, "Statistical shape analysis in high-level vision," *Mathematical Methods in Computer Vision*, p. 37, 2003.

[58] I. L. Dryden and K. V. Mardia, *Statistical Shape Analysis*. Wiley New York, 1998.

[59] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM Journal on Matrix Analysis and Application*, vol. 20, no. 2, pp. 303–353, 1999.

[60] G. J. Edwards, T. F. Cootes, and C. J. Taylor, "Face recognition using active appearance models," in *Proceedings of the European Conference on Computer Vision*, vol. 2, pp. 581–695, 1998.

[61] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1151–1163, 2002.

[62] A. Elgammal and C. Lee, "Separating style and content on a nonlinear manifold," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 1, 2004.

[63] A. Elgammal and C. S. Lee, "Inferring 3D body pose from silhouettes using activity manifold learning," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 681–682, 2004.

[64] A. Erdelyi, *Asymptotic Expansions*. Dover, 1956.

[65] O. Faugeras, Q. T. Luong, and T. Papadopoulo, *The Geometry of Multiple Images*. Massachusetts: MIT Press Cambridge, 2001.

[66] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 55–79, 2005.

[67] H. Fillbrandt and K. H. Kraiss, "Tracking people on the ground plane of a cluttered scene with a single camera," *WSEAS Transactions on Information Science and Applications*, vol. 2, pp. 1302–1311, 2005.

[68] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[69] F. Fleuret, J. Berclaz, and R. Lengagne, "Multi-camera people tracking with a probabilistic occupancy map," Technical Report, EPFL/CVLAB2006.07, July 2006.

[70] A. Forner-Cordero, H. Koopman, and F. van der Helm, "Describing gait as a sequence of states," *Journal of Biomechanics*, vol. 39, no. 5, pp. 948–957, 2006.

[71] G. D. Forney Jr, "The Viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[72] W. Forstner, "Uncertainty and projective geometry," *Handbook of Geometric Computing: Applications in Pattern Recognition, Computer Vision, Neuralcomputing and Robotics*, pp. 493–534, 2005.

[73] H. Freeman, "On the encoding of arbitrary geometric configurations," *IRE Transactions on Electronic Computers*, vol. 10, no. 2, pp. 260–268, 1961.

[74] D. M. Gavrila and L. S. Davis, "3-D model-based tracking of humans in action: A multi-view approach," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 73–80, 1996.

[75] D. Geiger, T. Liu, and R. V. Kohn, "Representation and self-similarity of shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 1, no. 1, pp. 86–99, 2003.

[76] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721–741, 1984.

[77] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, no. 6, pp. 643–660, 2001.

[78] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice Hall, 2007.

[79] C. R. Goodall and K. V. Mardia, "Projective shape analysis," *Journal of Computational and Graphical Statistics*, vol. 8, no. 2, pp. 143–168, 1999.

[80] N. Gordon, D. Salmon, and A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proceedings on Radar and Signal Processing*, vol. 140, pp. 107–113, 1993.

[81] A. Goshtasby, "Description and discrimination of planar shapes using shape matrices," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 7, no. 6, pp. 738–743, 1985.

[82] U. Grenander, *Abstract Inference*. John Wiley & Sons, 1981.

[83] U. Grenander, *General Pattern Theory: A Mathematical Study of Regular Structures*. USA: Oxford University Press, 1993.

[84] U. Grenander and M. I. Miller, *Pattern Theory: From Representation to Inference*. USA: Oxford University Press, 2007.

[85] U. Grenander and M. Rosenblatt, *Statistical Analysis of Stationary Time Series*. Chelsea Publishing Company, Incorporated, 1984.

[86] G. D. Hager and P. N. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 10, pp. 1025–1039, 1998.

[87] R. M. Haralick, "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 786–804, 1979.

[88] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the Alvey Vision Conference*, vol. 15, pp. 147–151, 1988.

[89] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[90] R. I. Hartley and P. Sturm, "Triangulation," *Computer Vision and Image Understanding*, vol. 68, no. 2, pp. 146–157, 1997.

[91] R. Horaud, F. Dornaika, and B. Lamiroy, "Object pose: The link between weak perspective, paraperspective, and full perspective," *International Journal of Computer Vision*, vol. 22, no. 2, pp. 173–189, 1997.

[92] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1–3, pp. 185–203, 1981.

[93] M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962.

[94] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 34, no. 3, pp. 334–352, 2004.

[95] P. J. Huber, *Robust Statistics*. Wiley New York, 1981.

[96] A. Hyvrinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. John Wiley and Sons, 2001.

[97] M. Isard and A. Blake, "Condensation — conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[98] M. Isard and A. Blake, "ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework," in *Proceedings of European Conference on Computer Vision*, vol. 1, pp. 767–781, 1998.

[99] A. K. Jain, *Fundamentals of Digital Image Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.

[100] T. S. Jebara and A. Pentland, "Parametrized structure from motion for 3D adaptive feedback tracking of faces," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 144–150, 1997.

[101] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1296–1311, 2003.

[102] S. Joo and Q. Zheng, "A temporal variance-based moving target detector," in *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, 2005.

[103] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Learning in graphical models*, pp. 105–161, 1999.

[104] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pp. 477–481, 2000.

[105] S. Julier and J. K. Uhlmann, "A general method for approximating nonlinear transformations of probability distributions," Technical Report, Department of Engineering Science, University of Oxford, 1996.

[106] A. Kale, A. N. Rajagopalan, A. Sundaresan, N. Cuntoor, A. Roy Cowdhury, V. Krueger, and R. Chellappa, "Identification of Humans using gait," *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1163–1173, 2004.

[107] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME Journal of Basic Engineering*, vol. 82D, no. 1, pp. 34–45, 1960.

[108] K. Kanatani, *Group Theoretical Methods in Image Understanding*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1990.

[109] K. Kanatani, *Statistical Optimization for Geometric Computation: Theory and Practice*. New York, NY, USA: Elsevier Science Inc., 1996.

[110] R. Kashyap and R. Chellappa, "Stochastic models for closed boundary analysis: Representation and reconstruction," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 627–637, 1981.

[111] R. Kashyap and R. Chellappa, "Estimation and choice of neighbors in spatial-interaction models of images," *IEEE Transactions on Information Theory*, vol. 29, no. 1, pp. 60–72, 1983.

[112] R. L. Kashyap, "Analysis and synthesis of image patterns by spatial interaction models," *Progress in Pattern Recognition*, vol. 1, pp. 149–186, 1981.

[113] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 506–513, 2004.

[114] D. G. Kendall, "Shape manifolds, procrustean metrics, and complex projective spaces," *Bulletin of the London Mathematical Society*, vol. 16, no. 2, pp. 81–121, 1984.

[115] S. M. Khan and M. Shah, "A multiview approach to tracking people in crowded scenes using a planar homography constraint," in *Proceedings of European Conference on Computer Vision*, vol. 4, pp. 133–146, 2006.

[116] Z. Khan, T. Balch, and F. Dellaert, "An MCMC-based particle filter for tracking multiple interacting targets," in *Proceedings of European Conference on Computer Vision*, vol. 4, pp. 279–290, 2004.

[117] A. Khotanzad and Y. H. Hong, "Invariant image recognition by Zernike moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 5, pp. 489–497, 1990.

[118] K. Kim and L. S. Davis, "Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering," in *Proceedings of European Conference on Computer Vision*, vol. 3, pp. 98–109, 2006.

[119] S. Kirkpatrick, "Optimization by simulated annealing: Quantitative studies," *Journal of Statistical Physics*, vol. 34, no. 5, pp. 975–986, 1984.

[120] O. Kouropteva, O. Okun, and M. Pietikainen, "Classification of handwritten digits using supervised locally linear embedding algorithm and support vector machine," in *Proceedings of 11th European Symposium Artificial Neural Networks*, pp. 229–234, 2003.

[121] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[122] M. La Cascia, S. Sclaroff, and V. Athitsos, "Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3D models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 4, pp. 322–336, 2000.

[123] J. H. Lambert, *Photometria Sive de Mensure de Gratibus Luminis, Colorum Umbrae*. Eberhard Klett, 1760.

[124] S. L. Lauritzen, *Graphical Models*. Oxford University Press, 1996.

[125] K. C. Lee, J. Ho, M. H. Yang, and D. Kriegman, "Video-based face recognition using probabilistic appearance manifolds," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 313–320, 2003.

[126] F. F. Li and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 524–531, 2005.

[127] J. Li and R. Chellappa, "Structure from planar motion," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3466–3477, 2006.

[128] H. Ling and D. W. Jacobs, "Deformation invariant image matching," in *Proceedings of IEEE International Conference on Computer Vision*, vol. 2, pp. 1466–1473, 2005.

[129] H. Ling and D. W. Jacobs, "Shape classification using the inner-distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 286–299, 2007.

[130] J. S. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *Journal of American Statistician Association*, vol. 93, no. 443, pp. 1032–1044, 1998.

[131] J. S. Liu, R. Chen, and T. Logvinenko, "A theoretical framework for sequential importance sampling with resampling," in *Sequential Monte Carlo Methods in Practice*, (A. Doucet, N. de Freitas, and N. Gordon, eds.), New York: Springer-Verlag, 2001.

[132] Z. Liu and S. Sarkar, "Improved gait recognition by gait dynamics normalization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 6, pp. 863–876, 2006.

[133] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, no. 1, pp. 133–135, 1981.

[134] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[135] F. Lv, T. Zhao, and R. Nevatia, "Camera calibration from video of a walking human," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, pp. 1513–1518, 2006.

[136] T. Lv, B. Ozer, and W. Wolf, "A real-time background subtraction method with camera motion compensation," in *Proceedings of IEEE International Conference on Multimedia and Expo*, vol. 1, pp. 331–334, 2004.

[137] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer-Verlag, 2003.

[138] B. S. Manjunath, T. Simchony, and R. Chellappa, "Stochastic and deterministic networks for texture segmentation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 6, pp. 1039–1049, 1990.

[139] J. Marrowuin, S. Mitter, and T. Poggio, "Probabilistic solution of ill-posed problems in computational vision," *Journal of the American Statistical Association*, vol. 82, no. 397, pp. 76–89, 1987.

[140] P. Meer, D. Mintz, A. Rosenfeld, and D. Y. Kim, "Robust regression methods for computer vision: A review," *International Journal of Computer Vision*, vol. 6, no. 1, pp. 59–70, 1991.

[141] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equations of state calculations by fast computing machines," *Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1091, 1953.

[142] W. Mio, A. Srivastava, and S. H. Joshi, "On Shape of Plane Elastic Curves," *International Journal of Computer Vision*, vol. 73, no. 3, pp. 307–324, 2007.

[143] A. Mittal and L. S. Davis, "M2 Tracker: A multi-view approach to segmenting and tracking people in a cluttered scene," *International Journal of Computer Vision*, vol. 51, no. 3, pp. 189–203, 2003.

[144] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 696–710, 1997.

[145] F. Mura and N. Franceschini, "Visual control of altitude and speed in a flying agent," in *From Animals to Animats 3: Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, pp. 91–99, MIT Press, 1994.

[146] T. R. Neumann and H. H. Bulthoff, "Insect inspired visual control of translatory flight," in *Proceedings of the 6th European Conference on Advances in Artificial Life*, pp. 627–636, 2001.

[147] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, 2004.

[148] B. Ochoa and S. Belongie, "Covariance propagation for guided matching," in *Proceedings of Workshop on Statistical Methods in Multi-Image and Video Processing (SMVP)*, 2006.

[149] J. Oliensis, "A critique of structure from motion algorithms," Technical Report, NEC Research Institute, 2000.

[150] S. K. Parui, S. E. Sarma, and D. D. Majumder, "How to discriminate shapes using shape vector," *Pattern Recognition Letters*, vol. 4, no. 3, pp. 201–204, 1986.

[151] V. Patrangenaru and K. V. Mardia, "Affine shape analysis and image analysis," in *22nd Leeds Annual Statistics Research Workshop*, 2003.

[152] T. Pavlidis, "A review of algorithms for shape analysis," *Document Image Analysis*, pp. 145–160, 1995.

[153] E. Persoon and K. Fu, "Shape discrimination using Fourier descriptors," *IEEE Transactions on Man Machine and Cybernetics*, vol. 7, no. 3, pp. 170–179, 1977.

[154] J. Phillips, S. Sarkar, I. Robledo, P. Grother, and K. Bowyer, "The gait identification challenge problem: Data sets and baseline algorithm," in *Proceedings of IEEE International Conference on Pattern Recognition*, August 2002.

[155] B. T. Phong, "Illumination for computer generated pictures," *Communications of the ACM*, vol. 18, no. 6, pp. 311–317, 1975.

[156] C. J. Poelman and T. Kanade, "A paraperspective factorization method for shape and motion recovery," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, pp. 206–218, 1997.

[157] F. Porikli, O. Tuzel, and P. Meer, "Covariance tracking using model update based on lie algebra," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 728–735, 2006.

[158] M. J. Prentice and K. V. Mardia, "Shape changes in the plane for landmark data," *The Annals of Statistics*, vol. 23, no. 6, pp. 1960–1974, 1995.

[159] G. Qian and R. Chellappa, "Structure from motion using sequential Monte Carlo methods," *International Journal of Computer Vision*, vol. 59, no. 1, pp. 5–31, 2004.

[160] G. Qian and R. Chellappa, "Structure from motion using sequential Monte Carlo methods," *International Journal of Computer Vision*, vol. 59, no. 1, pp. 5–31, 2004.

[161] G. Qian, R. Chellappa, and Q. Zheng, "Spatial self-calibration of distributed cameras," in *Proceedings of Collaborative Technology Alliances Conference — Sensors*, 2003.

[162] L. Rabiner and B. Juang, "An Introduction to Hidden Markov Models," *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[163] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[164] L. R. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[165] A. Rangarajan, H. Chui, and F. L. Bookstein, "The softassign Procrustes matching algorithm," in *Proceedings of the International Conference on Information Processing in Medical Imaging*, pp. 29–42, 1997.

[166] C. R. Rao, *Linear Statistical Inference and its Applications*. John Wiley & Sons, 1973.

[167] H. Robbins and S. Monro, "A stochastic approximation method," *Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, 1951.

[168] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*. New York: Springer-Verlag, 1999.

[169] Y. A. Rosanov, "On Gaussian Fields with given conditional distributions," *Theory of Probability and its Applications*, vol. 12, no. 3, pp. 381–391, 1967.

[170] P. J. Rousseeuw, "Least median of squares regression," *Journal of American Statistical Association*, vol. 79, no. 388, pp. 871–880, 1984.

[171] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[172] A. K. Roy-Chowdhury and R. Chellappa, "Statistical bias in 3-D reconstruction from a monocular video," *IEEE Transactions on Image Processing*, vol. 14, no. 8, pp. 1057–1062, 2005.

[173] H. Rubin, "Robust Bayesian estimation," in *Proceedings of symposium on Statistical Decision Theory and Related Topics II*, pp. 351–356, New York: Academic Press, 1977.

[174] S. Saha, C. C. Shen, C. J. Hsu, G. Aggarwal, A. Veeraraghavan, A. Sussman, and S. S. Bhattacharyya, "Model-Based OpenMP implementation of a 3D facial pose tracking system," *International Workshop on Parallel Processing*, pp. 66–73, 2006.

[175] P. Saisan, G. Doretto, Y. N. Wu, and S. Soatto, "Dynamic texture recognition," in *Proceedings of IEEE International Conference on Computer Vision*, vol. 2, pp. 58–63, 2001.

[176] A. Sankaranarayanan, A. Veeraraghavan, and R. Chellappa, "Hybrid subspace sparse signals and their application to reflectance measurement," Manuscript under preparation.

[177] A. C. Sankaranarayanan and R. Chellappa, "Optimal multi-view fusion of object locations," in *Proceedings of IEEE Workshop on Motion and Video Computing (WMVC)*, pp. 1–8, 2008.

[178] A. C. Sankaranarayanan, J. Li, and R. Chellappa, "Fingerprinting vehicles for tracking across non-overlapping views," in *Army Science Conference*, 2006.

[179] A. C. Sankaranarayanan, A. Veeraraghavan, and R. Chellappa, "Object detection, tracking and recognition for multiple smart cameras," *Proceedings of the IEEE*, vol. 96, no. 10, pp. 1606–1624, 2008.

[180] S. Sarkar, P. Phillips, Z. Liu, I. R. Vega, P. Grother, and K. W. Bowyer, "The humanID gait challenge problem: Data sets, performance, and analysis," *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 2, pp. 162–177, 2005.

[181] Y. Y. Schechner, S. K. Nayar, and P. N. Belhumeur, "Multiplexing for optimal lighting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 8, pp. 1339–1354, 2007.

[182] J. Shao, S. K. Zhou, and R. Chellappa, "Video mensuration using stationary cameras," *IEEE Transactions on Image Processing* (under review).

[183] J. Shi and C. Tomasi, "Good features to track," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 593–600, 1994.

[184] J. Sivic, B. C. Russell, A. Efros, A. Zisserman, and W. T. Freeman, "Discovering object categories in image collections," in *Proceedings of IEEE International Conference on Computer Vision*, p. 65, 2005.

[185] L. I. Smith, *A Tutorial on Principal Components Analysis*. Cornell University, USA, 2002.

[186] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," *Autonomous Robot Vehicles*, pp. 167–193, 1990.

[187] R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *The International Journal of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1986.

[188] S. Soatto, G. Doretto, and Y. Wu, "Dynamic textures," in *Proceedings of IEEE International Conference on Computer Vision*, pp. 439–446, 2001.

[189] H. Sorenson, *Parameter Estimation: Principles and Problems*. M. Dekker, 1980.

[190] G. Sparr, "Depth computations from polyhedral images," *Image and Vision Computing*, vol. 10, no. 10, pp. 683–688, 1992.

[191] M. Srinivasan, S. Zhang, M. Lehrer, and T. Collett, "Honeybee navigation en route to the goal: Visual flight control and odometry," *Journal of Experimental Biology*, vol. 199, no. 1, pp. 237–244, 1996.

[192] S. Srinivasan, "Extracting structure from optical flow using the fast error search technique," *International Journal of Computer Vision*, vol. 37, no. 3, pp. 203–230, 2000.

[193] A. Srivastava, S. H. Joshi, W. Mio, and X. Liu, "Statistical shape analysis: Clustering, learning, and testing," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 27, no. 4, pp. 590–602, 2005.

[194] A. Srivastava and X. Liu, "Tools for application-driven linear dimension reduction," *Neurocomputing*, vol. 67, pp. 136–160, 2005.

[195] A. Srivastava, W. Mio, E. Klassen, and S. Joshi, "Geometric analysis of continuous, planar shapes," in *Proceedings of 4th International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pp. 341–356, 2003.

[196] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, 2000.

[197] J. Stuelpnagel, "On the parametrization of the three-dimensional rotation group," *SIAM Review*, vol. 6, no. 4, pp. 422–430, 1964.

[198] Z. Sun, A. M. Tekalp, and V. Ramesh, "Error characterization of the factorization method," *Computer Vision and Image Understanding*, vol. 82, no. 2, pp. 110–137, 2001.

[199] A. Sundaresan and R. Chellappa, "Model driven segmentation of articulating humans in Laplacian Eigenspace," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1771–1785, 2008.

[200] D. Takhar, J. N. Laska, M. B. Wakin, M. F. Duarte, D. Baron, S. Sarvotham, K. F. Kelly, and R. G. Baraniuk, "A New Compressive Imaging Camera Architecture using Optical-Domain Compression," in *Proceedings of the SPIE Computational Imaging IV*, vol. 6065, pp. 43–52, 2006.

[201] J. B. Tenenbaum, V. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[202] C. Tomasi and T. Kanade, "Detection and tracking of point features," Technical Report, CMU-CS-91–132, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1991.

[203] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization method," *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137–154, November 1992.

[204] B. Triggs, "Factorization methods for projective structure and motion," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 845–851, 1996.

[205] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment — A modern synthesis," in *Proceedings of the International Workshop on Vision Algorithms*, pp. 298–372, 2000.

[206] P. Turaga and R. Chellappa, "Locally time-invariant models of human activities using trajectories on the Grassmannian," in *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 2435–2441, 2009.

[207] P. Turaga, A. Veeraraghavan, and R. Chellappa, "Statistical analysis on Stiefel and Grassmann manifolds with applications in computer vision," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[208] P. K. Turaga, A. Veeraraghavan, and R. Chellappa, "From videos to verbs: Mining videos for events using a cascade of dynamical systems," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2007.

[209] P. K. Turaga, A. Veeraraghavan, and R. Chellappa, "Unsupervised view and rate invariant clustering of video sequences," *Computer Vision and Image Understanding*, vol. 113, no. 3, pp. 353–371, 2009.

[210] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[211] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 586–591, 1991.

[212] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification," in *European Conference on Computer Vision*, pp. 589–600, 2006.

[213] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian detection via classification on Riemannian manifolds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1713–1727, 2008.

[214] A. Veeraraghavan, R. Chellappa, and A. K. Roy-Chowdhury, "The function space of an activity," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 959–968, 2006.

[215] A. Veeraraghavan, R. Chellappa, and M. Srinivasan, "Shape-and-behavior encoded tracking of bee dances," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 3, pp. 463–476, 2008.

[216] A. Veeraraghavan, D. Reddy, and R. Raskar, "Coded strobing camera for high speed periodic events," *IEEE Transactions on Pattern Analysis and Machine Intelligence* (under revision), 2009.

[217] A. Veeraraghavan, A. K. Roy-Chowdhury, and R. Chellappa, "Matching shape sequences in video with applications in human movement analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1896–1909, 2005.

[218] A. Veeraraghavan, A. Srivastava, A. K. Roy-Chowdhury, and R. Chellappa, "Rate-invariant recognition of humans and their activities," *IEEE Transactions on Image Processing*, vol. 18, no. 6, pp. 1326–1339, 2009.

[219] K. von Frisch, *The Dance Language and Orientation of Bees*. Cambridge, Massachusetts: Belknap Press, 1967.

[220] Y. Wang, H. Jiang, M. S. Drew, Z. N. Li, and G. Mori, "Unsupervised discovery of action classes," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1654–1661, 2006.

[221] D. Weinland, R. Ronfard, and E. Boyer, "Free viewpoint action recognition using motion history volumes," *Computer Vision and Image Understanding*, vol. 104, no. 2, pp. 249–257, 2006.

[222] P. Whittle, "On stationary processes in the plane," *Biometrika*, vol. 41, no. 3–4, pp. 434–449, 1954.

[223] J. Woods, "Two-dimensional discrete Markovian fields," *IEEE Transactions on Information Theory*, vol. 18, no. 2, pp. 232–240, 1972.

[224] C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780–785, 1997.

[225] M. Xu, J. Orwell, and G. Jones, "Tracking football players with multiple cameras," in *Proceedings of IEEE International Conference on Image Processing*, pp. 2909–2912, 2004.

[226] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computing Surveys (CSUR)*, vol. 38, no. 4, pp. 1–45, 2006.

[227] G.-S. J. Young and R. Chellappa, "Statistical analysis of inherent ambiguities in recovering 3-D motion from a noisy flow field," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 10, pp. 995–1013, 1992.

[228] J. Zhang, S. Z. Li, and J. Wang, "Nearest manifold approach for face recognition," in *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 17–19, 2004.

[229] Z. Zhang, "Parameter estimation techniques: A tutorial with application to conic fitting," *Image and Vision Computing*, vol. 15, no. 1, pp. 59–76, 1997.

[230] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimensionality reduction via tangent space alignment," *SIAM Journal on Scientific Computing*, vol. 26, no. 1, pp. 313–338, 2005.

[231] T. Zhao and R. Nevatia, "Tracking multiple humans in complex situations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1208–1221, 2004.

[232] T. Zhao and R. Nevatia, "Tracking multiple humans in crowded environment," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 406–413, 2004.

[233] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Computing Surveys*, vol. 35, no. 4, pp. 399–458, 2003.

[234] S. Zhou, V. Krueger, and R. Chellappa, "Probabilistic recognition of human faces from video," *Computer Vision and Image Understanding*, vol. 91, no. 1–2, pp. 214–245, 2003.

[235] S. K. Zhou, R. Chellappa, and B. Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters," *IEEE Transactions on Image Processing*, vol. 13, no. 11, pp. 1491–1506, 2004.