P

## Panoramic Camera

►Omnidirectional Camera

## Panoramic Image Generation

►Video Mosaicing

## Panoramic Stitching

►Image Stitching

## Pan-Tilt Camera Calibration

►Active Calibration

## Pan-Tilt-Zoom (PTZ) Camera

Sudipta N. Sinha
Microsoft Research, Redmond, WA, USA

### Synonyms

IP camera; Network camera; Surveillance camera

### Related Concepts

►Camera Calibration

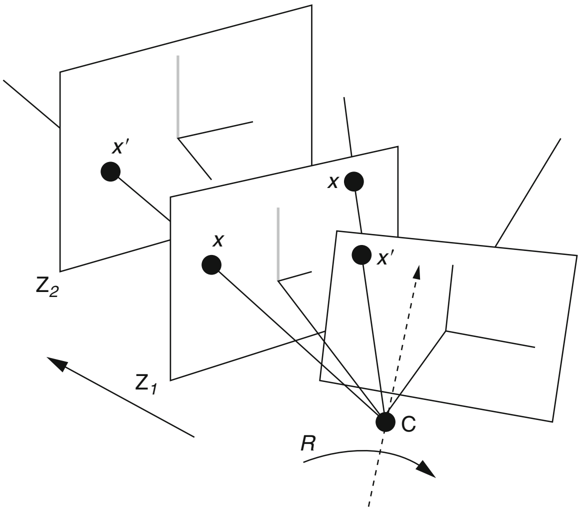
### Definition

A pan-tilt-zoom (PTZ) camera typically refers to an active camera which has some degree of pan, tilt, and zoom control. They are commonly used to monitor large areas for visual surveillance applications. The pan, tilt, and zoom controls of most off-the-shelf PTZ cameras are often programmable, enabling the camera to be remotely controlled over a network. Some variants of PTZ cameras are called *network* cameras when they come equipped with a real-time operation system which makes it possible to stream video over a network in real time. An example of such a camera is the Canon VB-C60 (see http://www.usa.canon.com/app/pdf/nvideo/VB-C60_Product_Sheet.pdf).

### Background

Active PTZ cameras serve as a practical alternative to high-resolution omnidirectional cameras in wide-area surveillance systems. On one hand, flexible pan and tilt ranges provide most PTZ cameras a large *effective* field of view (FOV) similar to an omnidirectional sensor, but in addition to that, these cameras can also zoom in on a small region of interest and capture it in high resolution. However, unlike omnidirectional sensors, PTZ cameras cannot simultaneously observe the complete scene. Since activities of interest in a visual surveillance scenario typically occur in small regions within a large area, simultaneous imaging is not always required. The inability to simultaneously observe the complete scene is often compensated by deploying a network of static and PTZ cameras.

K. Ikeuchi (ed.), *Computer Vision*, DOI 10.1007/978-0-387-31439-6,
© Springer Science+Business Media New York 2014



**Pan-Tilt-Zoom (PTZ) Camera, Fig. 1** 2D feature point correspondences under camera rotation and zoom

In order to utilize a programmable PTZ camera, many systems require the knowledge of a camera model that explains how 3D points in the world project to the image plane of a PTZ camera given a specific pan, tilt, and zoom configuration. Analyzing images from PTZ cameras may also require backprojecting a pixel on the image plane to obtain its corresponding 3D viewing ray (parameterized by the pan-tilt angles) on the camera's viewing sphere or the corresponding 3D ray in the world coordinate frame, provided the camera position is also known. For example, consider a calibrated PTZ camera monitoring a car parking area. For a calibrated camera, pixels can be mapped to precise locations on the ground plane. The model also makes it possible to actively track a person in the scene while ensuring that the PTZ camera is able to center its view on the individual. When a scene is monitored by a network of PTZ cameras, a calibrated camera model makes it possible to infer associations between objects or events detected in video captured from different viewpoints.

Conventional offline camera calibration methods such as [1] cannot easily be used to calibrate PTZ cameras since both the observed scene as well as the inter-camera baselines can be quite large. This precludes the use of conventional calibration objects which are typically too small for such large scenes [3, 4]. Also an active camera's calibration parameters must be continually estimated online by refining the pan-tilt angle and focal length estimates using a closed-loop mechanism [2] instead of relying on a set of precomputed pan-tilt control settings computed offline. This is due to the fact that the pan-tilt controllers present in most off-the-shelf PTZ cameras can be imprecise during operation.

## Theory

**PTZ Camera Model:** In many cases, PTZ cameras can be modeled using a simple motion model where the pan and tilt rotation axes are assumed to pass through the center of projection of the camera. However, depending on the camera's mechanical assembly and design, the pan and tilt rotation axes may need to be modeled as arbitrary axes not passing through the projection center [5]. In either case, the 2D projection of a 3D point onto the image plane can be computed using a few matrix operations. For indoor scenes, the intrinsic and extrinsic parameters of a network of pan-tilt cameras can be estimated offline by tracking a single moving LED over time within the working volume [5].

In outdoor scenes, the simple model works fine since the deviation of the rotation axes from the projection center is negligible compared to the average depth in the scene. This allows the PTZ camera to be modeled as a purely rotating and zooming camera for which self-calibration algorithms are well known [6]. A PTZ camera is then treated like a pinhole camera with a fixed projection center, but the camera intrinsics is modeled as a function of zoom and the camera's orientation obviously depends on the camera's pan and tilt settings.

Using homogeneous coordinates to represent a 3D world point and the corresponding image point denoted as $\mathbf{X}$ and $\mathbf{x}$, respectively, the imaging process can then be linearly modeled as follows:

$$\mathbf{x} = \mathbf{K}_z \mathbf{R}_{pt} [\mathbf{I} \,|\, -\mathbf{C}]\, \mathbf{X} \tag{1}$$

where $\mathbf{K}_z$ denotes the 3 × 3 camera *intrinsic* matrix, $\mathbf{R}_{pt}$ denotes the 3 × 3 rotation matrix, and $\mathbf{C}$ denotes the fixed projection center of the camera in world coordinates. The subscripts $z$, $p$, and $t$ refer to the zoom, pan, and tilt settings of the active camera. The rigid transformation from world coordinates to camera coordinates is defined by the rotation matrix $\mathbf{R}_{pt}$

and translation vector $\mathbf{t}_{pt} = -\mathbf{R}_{pt}\mathbf{C}$ which are often referred to as the camera's *extrinsic* parameters. $\mathbf{K}_z$ depends on the zoom setting $z$ and is defined in terms of five intrinsic parameters – $\alpha$, $s$, $f_z$, $u_z$, and $v_z$. $\alpha$ is the pixel aspect ratio (an unknown constant), $s$ is a skew parameter (typically set to 0), $f_z$ is the focal length measured in pixel, and $(u_z,v_z)$ is the principal point in the image plane:

$$\mathbf{K}_z = \begin{pmatrix} \alpha f_z & s & u_z \\ 0 & f_z & v_z \\ 0 & 0 & 1 \end{pmatrix} \qquad (2)$$

**Rotating and Zooming Cameras:** Let $\mathbf{x}$ and $\mathbf{x}'$ be the 2D projections of a 3D point $\mathbf{X}$ for two different PTZ configurations (see Fig. 1). Based on Eq. 1, $\mathbf{x} = \mathbf{K}[\mathbf{R}\ \mathbf{t}]\mathbf{X}$ and $\mathbf{x}' = \mathbf{K}'[\mathbf{R}'\ \ \mathbf{t}]\mathbf{X}$. Selecting $\mathbf{C}$ as the world origin results in $\mathbf{t} = \mathbf{0}$ and then eliminating $\mathbf{X}$ leads to the following:

$$\mathbf{x}' = \mathbf{K}'\mathbf{R}'\mathbf{R}^{-1}\mathbf{K}^{-1}\mathbf{x}$$

For rotation at constant zoom, the intrinsics are constant. Therefore,

$$\mathbf{x}' = \mathbf{K}\mathbf{R}_{\mathbf{rel}}\mathbf{K}^{-1}\mathbf{x} \qquad (3)$$

where $\mathbf{R}_{\mathbf{rel}} = \mathbf{R}'\mathbf{R}^{-1}$ denotes the relative 3D rotation about $\mathbf{C}$ between the two views and $\mathbf{K}$ denotes the intrinsic for the fixed zoom level. Similarly for a zooming camera with fixed orientation,

$$\mathbf{x}' = \mathbf{K}'\mathbf{K}^{-1}\mathbf{x} \qquad (4)$$

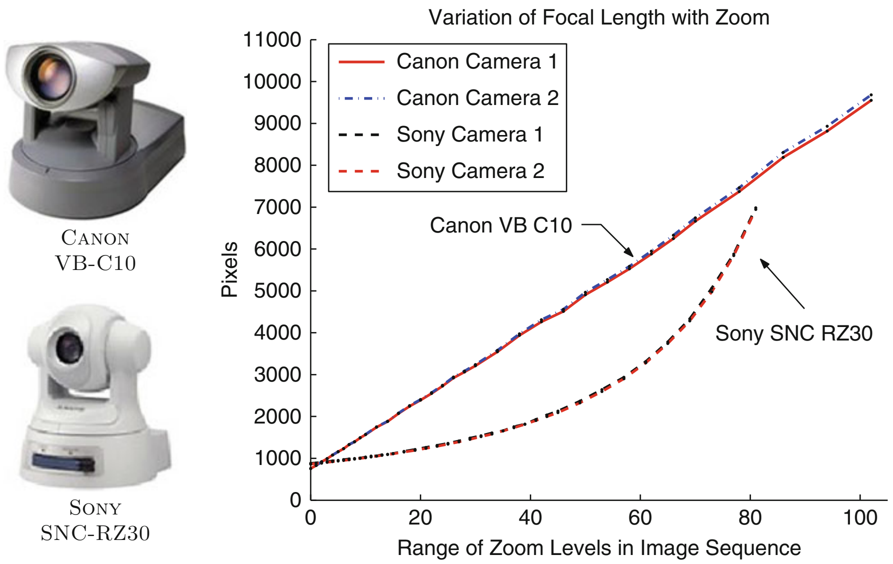
These 2D homographies $\mathbf{H}_{\mathbf{rot}} = \mathbf{K}\mathbf{R}_{\mathbf{rel}}\mathbf{K}^{-1}$ and $\mathbf{H}_{\mathbf{zoom}} = \mathbf{K}'\mathbf{K}^{-1}$ are used for automatic calibration of PTZ cameras [7, 8].

**Radial Distorting Correction:** PTZ cameras deviate from an ideal pinhole model due to radial distortion. This can be corrected using a standard model [9, 10]. The effect of radial distortion is more pronounced for smaller focal lengths, i.e., for lower zoom settings. The coefficients of radial distortion for a PTZ camera are therefore functions of zoom. The center of distortion is often assumed to coincide with the principal point $(u_z, v_z)$ which is also dependent on zoom [11].

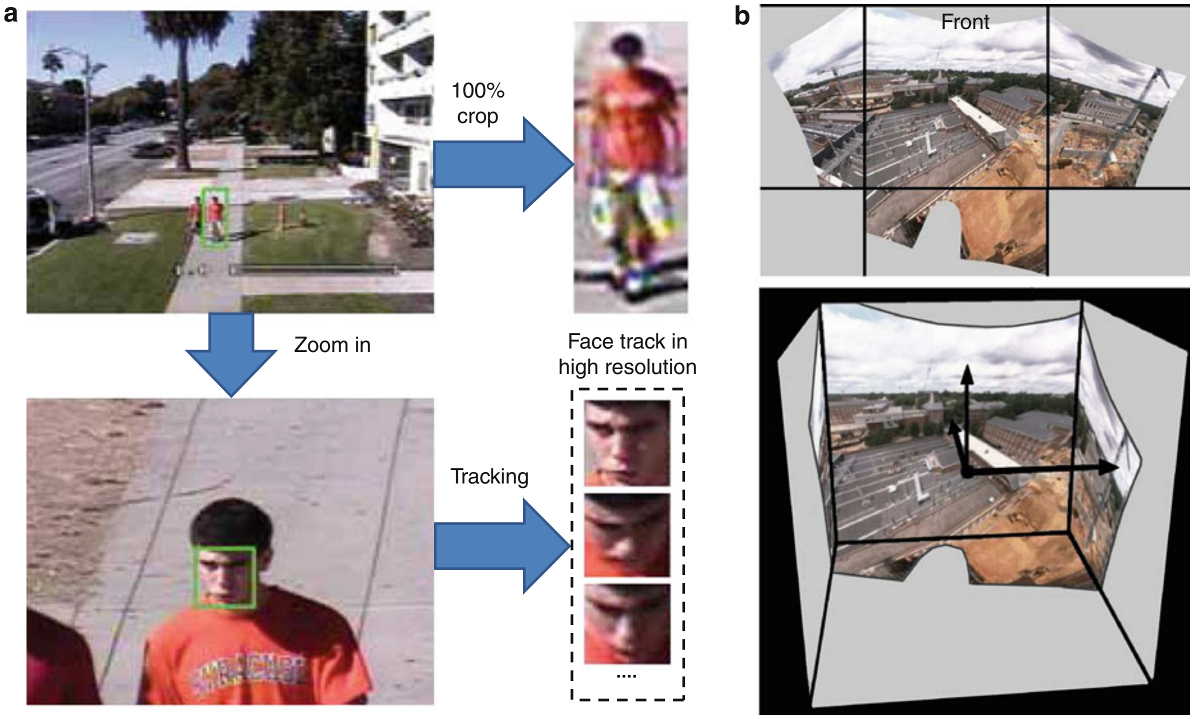
**PTZ Camera Calibration:** For a PTZ camera, the camera calibration process involves estimating the camera intrinsics $\mathbf{K}_z$ and coefficients of radial distortion for all zoom settings within the admissible range. Some other methods only model the variation of focal length with zoom [8]. In practice, the intrinsics are computed at a discrete set of zoom values, and during online operation the intrinsics for any zoom setting can be obtained by piecewise interpolation after looking up values in a precomputed table [7, 8]. The extrinsic parameters of a PTZ camera can be computed using image observations of 3D points with known coordinates [3, 10]. Alternatively, structure from motion techniques may also be used to calibrate a PTZ camera network where a common area is visible to all the cameras. The process is similar to the calibration of conventional camera networks [12, 13] but requires handling dynamic network topologies due to changing visibility relations between active cameras in the network [14].

To maintain precise calibration of an active PTZ camera in deployment, one must also address the fact that most PTZ cameras lack precise and repeatable pan-tilt controllers. This implies that even if the motorized controllers are precalibrated, the true 3D rotation of the camera for a particular pan-tilt setting of the camera may differ from the orientation predicted based on the precalibration. Systems where a precise calibration must be maintained may require a closed-loop calibration system depending on the accuracy of the PTZ controllers [7]. For closed-loop calibration, a calibrated panoramic mosaic must be computed offline. During online operation, this image serves as a calibration reference for all video frames captured online. Every video frame can then be aligned to the calibrated mosaic using a homography computed online with a feature-based method. This makes it possible to estimate the true 3D directions on the camera's viewing sphere for all pixels in the video stream of the active camera.

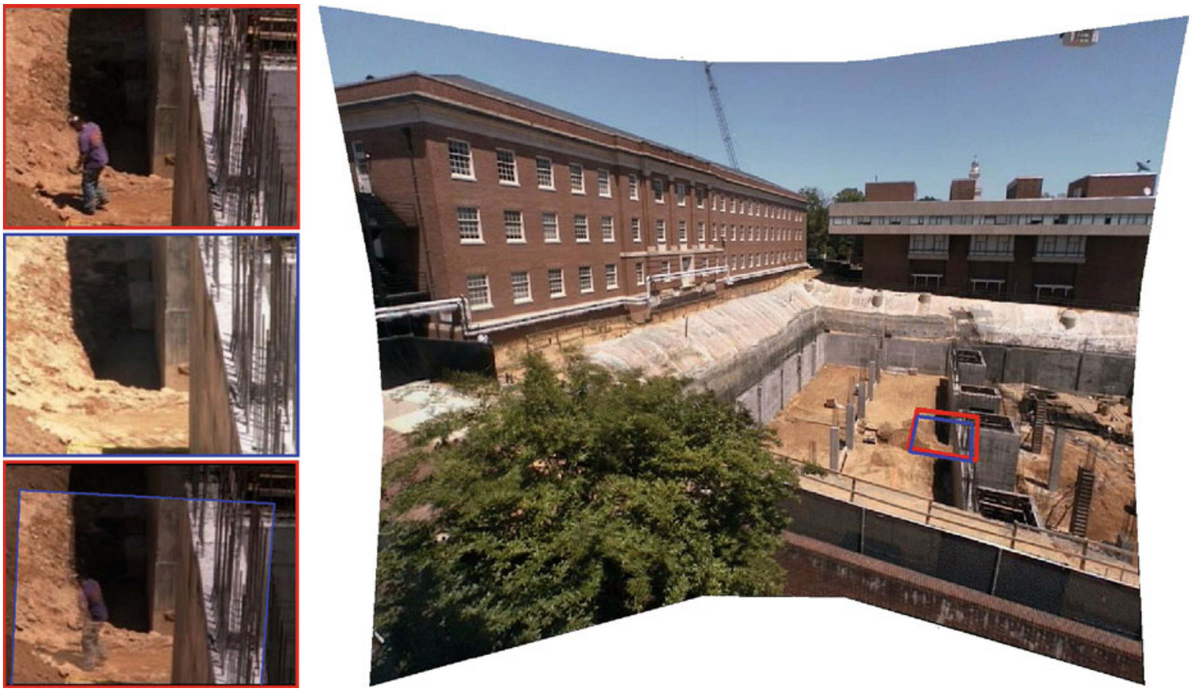
In [7], both camera intrinsics and the calibrated mosaic are computed automatically using the motion of natural scene features induced by actively panning, tilting, and zooming the camera. The 2D feature correspondences are robustly computed using the 2D homography-based motion model for rotating and zooming cameras. Fig. 2 shows parameters estimated using this method for two common off-the-shelf PTZ cameras. This method makes it possible to automatically recalibrate PTZ cameras after they are deployed on site.



**Pan-Tilt-Zoom (PTZ) Camera, Fig. 2** The variation of the focal length parameter with the zoom control for two common PTZ cameras. These parameters were estimated using an automatic method for PTZ camera calibration



**Pan-Tilt-Zoom (PTZ) Camera, Fig. 3** (**a**) Examples of high-resolution face images captured by an active PTZ camera tracking system is shown. These images were acquired after pedestrians were detected in the zoomed out view of the camera. (**b**) (*top*) A seamless panorama shown as an unwrapped cubemap generated by automatically stitching images from a rooftop PTZ surveillance camera is shown. (*bottom*) The calibrated cubemap is shown in the camera coordinate system



**Pan-Tilt-Zoom (PTZ) Camera, Fig. 4** Closed-loop control of an active PTZ camera. (*Left-top*) Video frame captured online. (*Left-middle*) Frame generated from the calibrated panorama based on predicted orientation. (*Left-bottom*) The aligned video frame. (*Right*) The predicted and aligned frames shown overlaid on the panorama

## Application

**Capturing High-Resolution Images of Humans:** Surveillance cameras monitoring large scenes cannot capture images of human faces at a sufficient resolution for identifying the individual when they are at a distance. PTZ cameras have been used to automatically zoom in on the face and capture a series of high-resolution face images once a pedestrian is detected (see Fig. 3a) [15]. This has applications in forensic video analysis. A multi-view video acquisition system utilizing a network of controllable PTZ cameras was also described in [16]. It was designed to track a person within an indoor scene while automatically adapting the pan-tilt camera controls to keep the individual centered in all the views. Such a system can be used for video conferencing [17] and event broadcasting.

**Detection and Tracking:** PTZ cameras are often deployed as part of a network of cameras for visual surveillance. In certain cases, static cameras provide a global view of the environment and are used primarily for detection and for actively steering the PTZ camera to regions of interest or to actively track moving targets. Such systems find applications in traffic monitoring, tracking pedestrians, security systems, or automatic event detection. To monitor more complex environments, smart surveillance systems can use multiple calibrated PTZ cameras working in cooperation to track multiple targets [18, 19]. Accurate calibration is required to correctly match targets tracked by different PTZ cameras and for handing off a moving target from one camera to another [20].

**PTZ Closed-Loop Calibration:** An open-loop calibration system for PTZ cameras that rely only on precomputed calibration of the PTZ controls will tend to be inaccurate during operation due to the imprecise nature of the PTZ controllers or due to vibrations or other sources of instability. To deal with this, a closed-loop calibration system should be used which is based on a calibrated panorama constructed offline (see Fig. 3b for an example). Figure 4 shows an outdoor panorama used for closed-loop control in [7]. Using feature-based image alignment, the video frames are robustly aligned to the calibrated panorama which allows pixels in the video stream to be precisely mapped to 3D directions on the camera's viewing sphere. When extrinsic parameters are also known, this provides the ability to perform 3D scene reasoning within a PTZ camera network.

## References

1. Zhang Z (1999) Flexible camera calibration by viewing a plane from unknown orientations. In: International conference on computer vision (ICCV 1999), Kerkyra, vol 1, pp 666–673
2. Wu Z, Radke R (2012) Keeping a pan-tilt-zoom camera calibrated. IEEE Trans Pattern Anal Mach Intell 99:1. PrePrints
3. Bouguet J (2000) Matlab camera calibration toolbox http://www.vision.caltech.edu/bouguetj/calib_doc/
4. Tsai R (1987) A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. Robotics and Automation, IEEE Journal of, 3(4):323–344
5. Davis J, Chen X (1999) Calibrating pan-tilt cameras in wide-area surveillance networks. In: Proceedings of the IEEE international conference on computer vision (ICCV), Kerkyra, pp 528–534
6. de Agapito L, Hartley R, Hayman E (1999) Linear self calibration of a rotating and zooming camera. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Ft. Collins, pp 15–21
7. Sinha S, Pollefeys M (2006) Pan-tilt-zoom camera calibration and high-resolution mosaic generation. Comput Vis Image Underst 103(3):170–183
8. Sankaranarayanan K, Davis JW (2010) Ptz camera modeling and panoramic view generation via focal plane mapping. In: Proceedings of the 10th Asian conference on computer vision – vol. Part II. ACCV, Queenstown, pp 580–593
9. Hartley R, Zisserman A (2005) Multiple view geometry in computer vision, vol 23. Cambridge University Press, New York
10. Collins R, Tsin Y (1999) Calibration of an outdoor active camera system. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Ft. Collins, pp 528–534
11. Willson R, Shafer S (1993) What is the center of the image? In: Proceedings of the IEEE international conference on computer vision (ICCV), Berlin, pp 670–678
12. Sinha SN, Pollefeys M, McMillan L (2004) Camera network calibration from dynamic silhouettes. CVPR 1(1):195–202
13. Svoboda T, Martinec D, Pajdla T (2005) A convenient multicamera self-calibration for virtual environments. Presence Teleoper. Virtual Environ. 14(4):407–422
14. Devarajan D, Radke RJ, Chung H (2006) Distributed metric calibration of ad hoc camera networks. ACM Trans Sens Netw 2(3):380–403
15. Dinh TB, Vo N, Medioni GG (2011) High resolution face sequences from a ptz network camera. In: Ninth IEEE international conference on automatic face and gesture recognition (FG 2011), Santa Barbara. IEEE, pp 531–538
16. Collins RT, Amidi O, Kanade T (2002) An active camera system for acquiring multi-view video. In: Proceedings of the international conference on image processing, Rochester, pp 517–520
17. Liao C, Liu Q, Kimber D, Chiu P, Foote J, Wilcox L (2003) Shared interactive video for teleconferencing. In: Proceedings of the Berkeley, CA, USA, pp 546–554
18. Everts I, Sebe N, Jones GA (2007) Cooperative object tracking with multiple ptz cameras. In: Proceedings of the 14th international conference on image analysis and processing, ICIAP '07, Modena, pp 323–330
19. Collins R, Lipton A, Fujiyoshi H, Kanade T (2001) Algorithms for cooperative multisensor surveillance. Proc IEEE 89(10):1456–1477
20. Qureshi F, Terzopoulos D (2006) Surveillance camera scheduling: a virtual vision approach. Multimed Syst 12(3):269–283
21. Sinha SN, Pollefeys M, Kim SJ (2004) High-resolution multiscale panoramic mosaics from pan-tilt-zoom cameras. In: Proceedings of the Fourth Indian conference on computer vision, graphics and image processing, Calcutta, pp 28–33
22. Stillman S, Tanawongsuwan R, Essa I (1998) A system for tracking and recognizing multiple people with multiple cameras. In: Proceedings of the second international conference on audio-vision based person authentication, Washington DC, pp 96–101
23. Starzyk W, Qureshi FZ (2011) Multi-tasking smart cameras for intelligent video surveillance systems. In: IEEE conference on advanced video and signal based surveillance, Klagenfurt, vol 1, 154–159

# Pan-Tilt-Zoom Camera Calibration

▶Active Calibration

# Parametric Curve

Bo Zheng
Computer Vision Laboratory, Institute of Industrial Science, The University of Tokyo, Meguro-ku, Tokyo, Japan

## Related Concepts

▶Algebraic Curve; ▶Parametric Surface; ▶Splines

## Definition

A *parametric curve* $S$ in 2-dimensional Euclidean space has the following form:

$$\begin{aligned} S(\cdot) : \mathbb{R} &\rightarrow \mathbb{R}^2 \\ t &\mapsto (x(t), y(t)), t \in [a, b] \end{aligned} \tag{1}$$

where $t$ is the parameter and varies in the domain $[a, b]$. In practice, the domain $[a, b]$ is often normalized as a specific region, such as $[0, 1]$. And $x(t)$, $y(t)$ are real-valued functions continuously mapping to a 2D point on a curve.

Similarly, a parametric curve $S$ in 3-dimensional space has the following form:

$$\begin{aligned} S(\cdot) &: \mathbb{R} \to \mathbb{R}^3 \\ t &\mapsto (x(t), y(t), z(t)), t \in [a, b] \end{aligned} \tag{2}$$

For example, an ellipse can be represented in a parametric form as: $x = a\cos t, y = b\sin t$, with $t \in [0, 2\pi)$, in contrast with its implicit representation: $\frac{x^2}{a^2} + \frac{y^2}{b^2} - 1 = 0$.

## Background

Since parametric functions are easy to construct and analyze, parametric curves are one of the most popular representations of general curves in computer graphics and computer vision. There are many types of parametric curves because of the flexibility in choosing the underlying analytic function. Spline curves are one of the most widely used parametric curves. Some popular spline curves are cubic curves, Hermite curves, Bézier curves, and B-spline curves (see *Splines* for further details).

Compared to nonparametric representations of curves, such as the implicit representation of *algebraic curves* and other explicit representation, parametric curves are easier to control the local geometry and thus attractive for interactive curve design or representing the nonrigid deformation; parametric curves are capable of approximating complex shapes with desired smoothness; parametric curves are independent of the choice of coordinate system and lend themselves well to geometric transformations.

## Application

In computer vision, parametric curves are generally used in modeling image edges, contours, and 2D object boundaries. It is shown as a powerful tool for shape fitting and manipulating. There are numerous applications, such as image segmentation, rigid/nonrigid object registration, motion estimation, object tracking, that benefit from parametric curve representations. For example, in the Snake-based image segmentation designed by Kass et al. [6] and modified by Brigger et al. [2], splines are used to model the image contours which converge to object shapes by minimizing the energy guided by external and internal forces. For image motion estimation, Szeliski and Coughlan [11] proposed to represent the local motion flow field using multi-resolution splines. And grid-based splines, such as Thin-Plate spline (TPS) [3] and Free Form Deformation (B-spline) proposed by Sederberg [9], can be effectively applied to nonrigid image registration. Other types of parametric curves are proposed for specific vision problems. For instance, Rational Gaussian curves [5] does not require a regular grid of control points and is suitable for shape recovery. And elastic strings [8] have metrics which are invariant under reparametrizations of curves and are useful for modeling elastic objects.

## References

1. Bartels RH, Beatty JC, Barsky BA (1987) An introduction to splines for use in computer graphics and geometric modeling. Morgan Kaufmann Publishers Inc., San Francisco
2. Brigger P, Hoeg J, Unser M (2000) B-spline snakes: a flexible tool for parametric contour detection. IEEE Trans Image Process 9(9):1484–1496
3. Duchon J (1977) Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In: Constructive theory of functions of several variables, lecture notes in mathematics, vol 571. Springer, Berlin, pp 85–100
4. Farin G (1990) Curves and surfaces for computer aided geometric design. Academic, San Diego
5. Goshtasby A (1992) Design and recovery of 2-D and 3-D shapes using rational gaussian. Int J Comput Vis 10(3): 233–256
6. Kass M, Witkin A, Terzopoulos D (1988) Snakes: active contour models. Int J Comput Vis 1:321–331
7. Kaufman A (1987) Efficient algorithms for 3D scan-conversion of parametric curves, surfaces, and volumes. In: Proceedings of the 14th annual conference on computer graphics and interactive techniques, Association for Computing Machinery, New York, pp 171–179
8. Mio W, Bowers JC, Liu X (2009) Shape of elastic strings in euclidean space. Int J Comput Vis 82(1):96–112
9. Sederberg T (1986) Free-form deformation of solid geometric models. ACM SIGGRAPH Comput Graph 20(4): 151–160
10. Sederberg TW, Anderson DC, Goldman RN (1984) Implicit representation of parametric curves and surfaces. Comput Vis Graph Image Process 28(1):72–84
11. Szeliski R, Coughlan J (1997) Spline-based image registration. Int J Comput Vis 22(3):199–218

## Parametric Surface

Bo Zheng
Computer Vision Laboratory, Institute of Industrial Science, The University of Tokyo, Meguro-ku, Tokyo, Japan

### Related Concepts

▶Algebraic Curve; ▶Parametric Curve; ▶Splines

### Definition

A parametric surface is a surface in the Euclidean space $\mathbb{R}^3$ which is defined by a parametric equation with two parameters,

$$\begin{aligned}\mathbb{S}(\cdot) : \mathbb{R}^2 &\to \mathbb{R}^3 \\ (u, v) &\mapsto (x(u, v), y(u, v), z(u, v)),\end{aligned} \tag{1}$$

where $u, v$ are the parameters and vary within a certain 2D domain in the parametric $uv$-plane. $x(u, v), y(u, v), z(u, v)$ are the real-valued functions continuously mapping to the points on a surface.

For example, Bézier surface is one of the most commonly used parametric surfaces (patch) defined as

$$P(u, v) = \sum_{i=0}^{m} \sum_{j=0}^{n} B_i^m(u) B_j^n(v) \mathbf{p}_{ij}, \tag{2}$$

where $\mathbf{p}_{ij}$ ($\in \mathbb{R}^3$) are the control points of Bézier spline surface. $B_i^n(\cdot) : \mathbb{R} \to \mathbb{R}$ are the basis functions determined by *Bernstein polynomials* of degree $n$ (see the detail in contribution *splines*).

Parameterization is an important process of deciding and defining the parameters necessary for modeling a parametric surface. For example, through a spherical parameterization, a unit sphere can be described as

$$\begin{aligned}x &= \cos\theta \cos\varphi \\ y &= \cos\theta \sin\varphi \\ z &= \sin\theta \\ -\frac{\pi}{2} &\le \theta < \frac{\pi}{2}, 0 \le \varphi < \pi.\end{aligned} \tag{3}$$

### Background

Parametric representation is the most general method to represent a surface, because it is capable of modeling a complex shape in a compact set of parameters and with desired smoothness. Surfaces that occur in two of the main theorems of vector calculus, Stokes' theorem and the divergence theorem, are frequently given in a parametric form. Parametric surfaces also provide a convenient way for computing the curvature and arc length of curves on the surface, surface area, differential geometric invariants such as the first and second fundamental forms, Gaussian, mean, and principal curvatures. Compared to implicit representation, parametric representation is more convenient for image rendering, due to the local points on a surface that can be fast and precisely determined. In addition, they may provide the potential for feature retrieving and surface modifying. Some typical parametric surfaces (patches) are spline-based parametric surfaces, such as bicubic patches, Bézier patches, and B-spline patches (see the contribution *splines*).

### Application

Parametric surface attracts attention in computer vision, because of its convenience in handling 3D image segmentation, nonrigid registration, and recognition. For example, a classic method for image segmentation, the level set method introduced by Osher and Fedkiw [8], employs parametric curve/surface to represent image contours (level sets) for tracking shapes. The parametric curve/surface makes it easy to handle shapes that change topology, e.g., shape splits in two, develops holes, or the reverse of these operations. The snake-based image segmentation designed by Kass et al. [7] and modified by Brigger et al. [2] employs splines to model the image contours and serving for an energy minimization guided by external and internal forces. Spherical harmonics represented in spherical parameterized coordinates play a special role in a wide variety of topics including indirect lighting and in recognition of 3D shapes [5]. And grid-based parametric surface, such as Thin-Plate spline (TPS) [3] and Free-Form Deformation (B-spline) proposed by Sederberg [9], can be effectively applied for nonrigid object registration.

## References

1. Bartels RH, Beatty JC, Barsky BA (1987) An introduction to splines for use in computer graphics and geometric modeling. Morgan Kaufmann, San Francisco
2. Brigger P, Hoeg J, Unser M (2000) B-spline snakes: a flexible tool for parametric contour detection. IEEE Trans Image Proc 9(9):1484–1496
3. Duchon J (1977) Splines minimizing rotation-invariant seminorms in Sobolev spaces. Constructive theory of functions of several variables. Lect Notes Math 571:85–100
4. Farin G (1990) Curves and surfaces for computer aided geometric design. Academic, San Diego
5. Funkhouser T, Min P, Kazhdan M, Chen J, Halderman A, Dobkin D, Jacobs D (2003) A search engine for 3D models. ACM Trans Graph 22(1):83–105
6. George P-L, Borouchaki H, Laug P (2000) Parametric surface meshing using a combined advancing-front generalized Delaunay approach. Int J Numer Method Eng 49(1–2): 233–259
7. Kass M, Witkin A, Terzopoulos D (1988) Snakes: active contour models. Int J Comput Vis 1:321–331
8. Osher SJ, Fedkiw RP (2002) Level set methods and dynamic implicit surfaces. Springer, New York
9. Sederberg T (1986) Free-form deformation of solid geometric models. ACM SIGGRAPH Comput Graph 20(4):151–160

## Partitioning

▶Interactive Segmentation

## Penumbra and Umbra

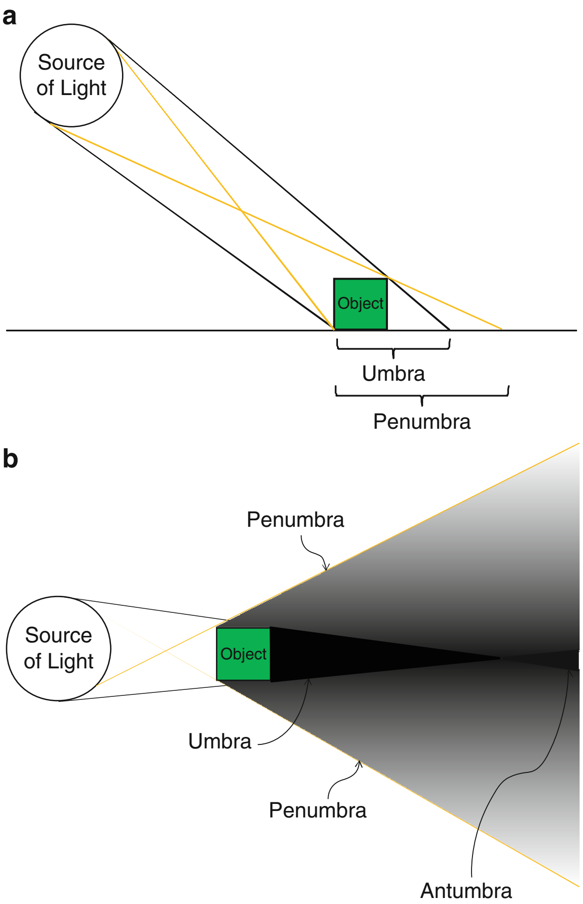
Rajeev Ramanath[1] and Mark S. Drew[2]
[1]DLP® Products, Texas Instruments Incorporated, Plano, TX, USA
[2]School of Computing Science, Simon Fraser University, Vancouver, BC, Canada

### Synonyms

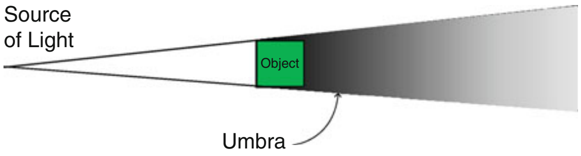
Shadow

### Definition

Umbra is the Latin name for "shadow" and is typically used to refer to the shadow cast by an object when illuminated by a light source [2]. In the umbra region of the shadow, the entire light source is occluded by the object. The region around the umbra where only a portion of the light is obscured is called the penumbra [1].



**Penumbra and Umbra, Fig. 1** (**a**) Shadows cast in the case of an object on a surface and (**b**) shadows cast in the case of an object in free-space



**Penumbra and Umbra, Fig. 2** Shadows cast in the case of an object in free space with a point source of light

Considering a simplistic arrangement of a light source and an object, Fig. 1a shows the regions denoted as the umbra and penumbra. As well, Fig. 1b shows the antumbra, the region where the light source actually appears bigger than the object; an antumbra is typically formed only when the object is in free-space, as opposed to resting on a surface.

The above considerations are for an area light source. In contrast, it is straightforward to see that for a point source of light, only an umbra is cast (see Fig. 2).

In astronomical setups – similar to when objects are in free space – observers who see partial eclipses (of the sun or the moon) are located in the penumbra.

## References

1. Encyclopedia Britanicca (2010) http://www.britannica.com/EBchecked/topic/450494/penumbra. Accessed 10 Sept 2010
2. Encyclopedia Britanicca (2010) http://www.britannica.com/EBchecked/topic/613811/umbra. Accessed 10 Sept 2010

# Performance Capture

▶Motion Capture

# Perspective Camera

Zhengyou Zhang
Microsoft Research, Redmond, WA, USA

## Synonyms

Pinhole camera

## Related Concepts

▶Camera Calibration; ▶Camera Parameters (Intrinsic, Extrinsic); ▶Calibration of Projective Cameras; ▶Depth Distortion; ▶Perspective Transformation

## Definition

A *perspective camera* is a mathematical model of an ideal pinhole camera that follows perspective projection.

## Background

A modern camera generally consists of an enclosed hollow with an opening (aperture) at one end for light to enter, a lens positioned in front of the opening, and a recording surface on the other end. In an ideal pinhole camera, the camera aperture is described as a point and no lenses are used to focus light. In that case, the camera can be modeled by a perspective transformation, thus also called *perspective camera*.

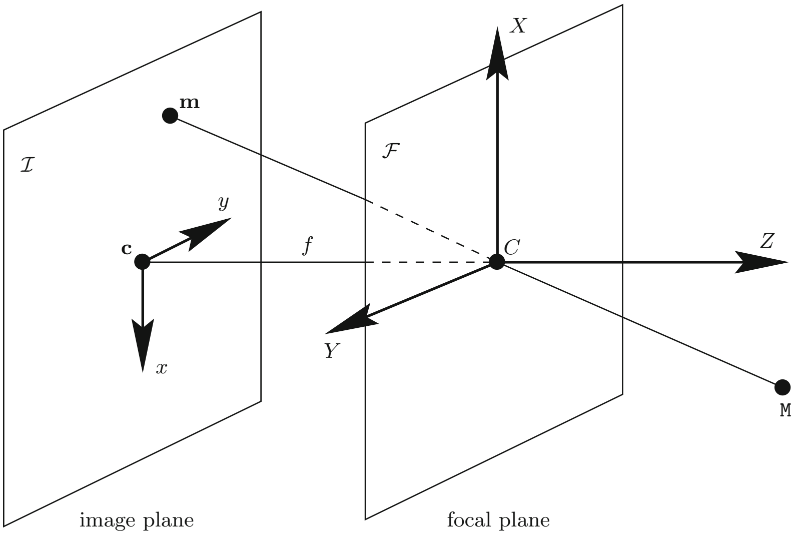
This model does not consider many effects of a real camera such as geometric distortions or blurring of unfocused objects caused by lenses and finite-sized apertures. Therefore, the pinhole camera model (perspective camera) can only be used as a first-order approximation of the transformation from a 3D scene to a 2D image. Its validity depends on the quality of the camera and the camera calibration process.

Lens distortion might be the major effect that the pinhole camera model does not take into account. It is sufficiently small if a high-quality camera is used and thus can be neglected. Otherwise, less distortion can be modeled through camera calibration and can be compensated for by applying suitable coordinate transformations on the image coordinates (lens distortion correction). Because of that, the pinhole camera model (perspective camera) has been the most popularly used camera model in computer vision and computer graphics to describe the relationship between a 3D scene and an image.

## Theory

Figure 1 shows a pinhole camera model. There is a plane $\mathcal{F}$ at a fixed distance $f$ in front of an *image plane* $\mathcal{I}$. The image plane is also called the *retinal plane*. An ideal pinhole $C$ is found in the plane $\mathcal{F}$. Assume that an enclosure is provided so that only light coming through the pinhole can reach the image plane. The rays of light emitted or reflected by an object pass through the pinhole and form an inverted image of that object on the image plane. Each point in the object, its corresponding image point, and the pinhole constitute a straight line. This kind of projection from 3D space to a plane is called *perspective projection*.

The geometric model of a pinhole camera thus consists of an image plane $\mathcal{I}$ and a point $C$ on the plane $\mathcal{F}$.



**Perspective Camera, Fig. 1** The pinhole camera model

The point $C$ is called the *optical center*, or the *focus*. The plane $\mathcal{F}$ going through $C$ and parallel to $\mathcal{I}$ is called the *focal plane*. The distance between the optical center and the image plane is the *focal length* of the optical system. The line going through the optical center $C$ and perpendicular to the image plane $\mathcal{I}$ is called the *optical axis*, and it intersects $\mathcal{I}$ at a point $c$, called the *principal point*. It is clear that the focal plane is also perpendicular to the optical axis. Experiences have shown that such a simple system can accurately model the geometry and optics of most of the modern Vidicon and CCD cameras [1].

Now let us derive the equations for the perspective projection. The coordinate system $(c, x, y)$ for the image plane is defined such that the origin is at the point $c$ (intersection of the optical axis with the image plane) and that the axes are determined by the camera scanning and sampling system. We choose the coordinate system $(C, X, Y, Z)$ for the three-dimensional space as indicated in Fig. 1, where the origin is at the optical center and the $Z$-axis coincides the optical axis of the camera. The $X$- and $Y$-axes are parallel, but opposite in direction, to the image $x$- and $y$-axes. The coordinate system $(C, X, Y, Z)$ is called the *standard coordinate system* of the camera, or simply *camera coordinate system*. From the above definition of the camera and image coordinate system, it is clear that the relationship between 2D image coordinates and 3D space coordinates can be written as

$$\frac{x}{X} = \frac{y}{Y} = \frac{f}{Z} \, . \tag{1}$$

It should be noted that, from the geometric viewpoint, there is no difference to replace the image plane by a virtual image plane located on the other side of the focal plane (Fig. 2). Actually this new system is what people usually use. In the new coordinate system, an image point $(x, y)$ has 3D coordinates $(x, y, f)$, if the scale of the image coordinate system is the same as that of the 3D coordinate system.

## Perspective Projection Matrix

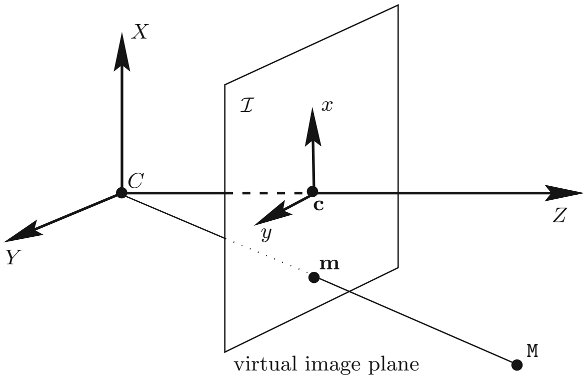
The relationship between 3D coordinates and image coordinates, Eq. (1), can be rewritten linearly as

$$\begin{bmatrix} U \\ V \\ S \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} , \tag{2}$$

where $x = U/S$ and $y = V/S$ if $S \neq 0$.

Given a vector $\mathbf{x} = [x, y, \cdots]^T$, we use $\widetilde{\mathbf{x}}$ to denote its augmented vector by adding 1 as the last element. Let $\mathbf{P}$ be the $3 \times 4$ matrix

$$\mathbf{P} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} ,$$



**Perspective Camera, Fig. 2** The pinhole camera model with a virtual image plane

which is called the camera *perspective projection matrix*. Given a 3D point $\mathbf{M} = [X, Y, Z]^T$ and its image $\mathbf{m} = [x, y]^T$, the formula (Eq. 2) can be written in matrix form as

$$s\widetilde{\mathbf{m}} = \mathbf{P}\widetilde{\mathbf{M}} \,, \tag{3}$$

where $s = S$ is an arbitrary nonzero scalar.

For a real image point, $S$ should not be 0. We now make an extension to include the case $S = 0$. If $S = 0$, then $Z = 0$, i.e., the 3D point is in the focal plane of the camera, and the image coordinates $x$ and $y$ are not defined. For all points in the focal plane but the optical center, their corresponding points in the image plane are at infinity. For the optical center $C$, we have $U = V = S = 0$ (i.e., $s = 0$) since $X = Y = Z = 0$.

The reader is referred to the entry *camera parameters (Intrinsic, Extrinsic)* for description of more general form of perspective projection matrix with intrinsic and extrinsic parameters [2].

## References

1. Faugeras O (1993) Three-dimensional computer vision: a geometric viewpoint. MIT, Cambridge
2. Xu G, Zhang Z (1996) Epipolar geometry in stereo, motion and object recognition. Kluwer Academic, Dordrecht/Boston

# Perspective Projection

▶Perspective Transformation

# Perspective Transformation

Zhengyou Zhang
Microsoft Research, Redmond, WA, USA

## Synonyms

Perspective camera; Perspective projection

## Related Concepts

▶Affine Camera; ▶Projection; ▶Weak Perspective Projection

## Definition

A *perspective transformation*, also called *perspective projection*, is a mathematical model to describe the projection performed by a perspective camera. Under perspective projection, an object that is closer to the camera appears larger than those farther away. See entry "▶Perspective Camera" for details.

# Phong Model

▶Phong Reflectance Model

# Phong Reflectance Model

Ping Tan
Department of Electrical and Computer Engineering,
National University of Singapore, Singapore,
Singapore

## Synonyms

Phong model

## Related Concepts

▶Radiance; ▶Reflectance Models; ▶Specularity, Specular Reflectance

## Definition

The Phong reflectance model calculates the amount of reflected radiance at a surface point according to the lighting, viewing, and surface normal directions at that point. It is characterized by modeling specular reflection as an exponential function of a cosine function, which provides moderate accuracy in a simple formulation. It was introduced by Bui Tuong Phong in 1973 in his PhD thesis and later published in [1].

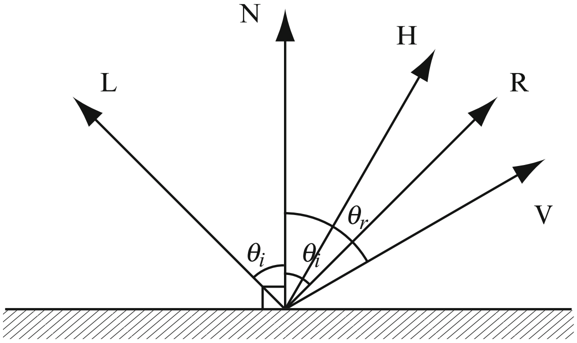
## Background

When light arrives at surfaces, it can be reflected, refracted, scattered, or absorbed. Reflectance models are mathematical functions that describe the interactions between light and surfaces. Usually, they are functions of lighting, viewing, and surface normal directions. There are various reflectance models at different levels of precision and complexity. Most reflectance models include components describing diffuse and specular reflection, respectively.

The Phong model is a reflectance model widely used for its simplicity and moderate accuracy. It represents reflected light as a linear combination of ambient, diffuse, and specular components. This model is characterized by its treatment of the specular component, which is designed based on the empirical observation that glossy surfaces have focused highlight that falls off quickly and matt surfaces have extended highlight that falls off slowly. Like many other reflectance models, this model can be applied to create computer graphic images or to infer scene properties such as shape and material from observed image intensities.

## Theory

Light reflected by a surface can be roughly divided into specular and diffuse reflection. The specular reflection, or highlight, refers to the light reflected directly at the surface without entering it. Specular reflection typically is concentrated within a small angle in space. In comparison, diffuse reflection refers to the light that enters the surface and is distributed more uniformly in all reflected directions.

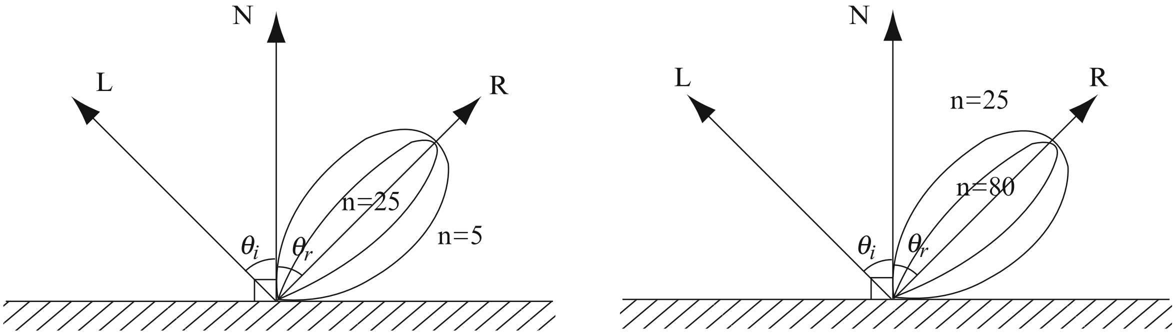


**Phong Reflectance Model, Fig. 1** Directions involved in the definition of Phong model. $N$ is the surface normal direction. $L$ and $V$ are the lighting and viewing direction, respectively. $H$ is the bisector of $L$ and $V$. $R$ is $L$ mirrored about $N$

The observed intensity of specular reflection depends on the viewing direction. For example, an ideal reflector like a mirror reflects light toward a direction $R$ that is the incident direction $L$ reflected about the surface normal direction $N$. Hence, the reflectance is zero except when the viewing direction $V$ is coincident with $R$. Here, $R$ is within the same plane as $L$ and $N$, and the angle between $R$ and $N$ is the same as that between $L$ and $N$. Mathematically, $R$ can be computed as $R = 2N(N \cdot L) - L$. All these directions are represented by unit vectors. The geometric relations of these unit vectors are shown in Fig. 1.

Most real surfaces are not ideal reflectors. The Phong model can be used to describe their specular reflection. According to this model, the specular reflection is centered at the direction $R$ and falls off as an exponential function of cosine of the angle between $R$ and $V$. In other words, the specular reflection is proportional to $(V \cdot R)^n$. $n > 0$ is known as shininess, which determines the spread of the specular reflection. It is larger for glossy surfaces and smaller for matt surfaces. Specular reflections with different shininess are illustrated on the left of Fig. 2. When $n = \infty$, it represents reflection of ideal reflectors. It should be noticed that Phong model is designed based on empirical observations rather than physics. There are other physically based reflectance models for specular reflection such as Cook-Torrance model [2], Ward's model [3], and Ashikhmin's model [4] that are more complicated and accurate than the Phong model.

The complete Phong reflectance model represents reflected light as a linear combination of ambient, diffuse, and specular components. The ambient component is a small constant accounting for weak scattered light in the scene. The diffuse component



**Phong Reflectance Model, Fig. 2** *Left*: the specular reflection of Phong model. *Right*: the specular reflection of Blinn-Phong model

is represented by the Lambert's model. Hence, the reflected radiance is computed as

$$I = k_a I_a + k_d I_d \max(N \cdot L, 0) + k_s I_s \left(\max(V \cdot R, 0)\right)^n . \quad (1)$$

Here, $I_a$, $I_d$, and $I_s$ are parameters to indicate the illumination power for the ambient, diffuse, and specular components. $k_a, k_d$, and $k_s$ are scalars that determine the relative strength of these three components. In color images, these parameters are vectors with different values at different wavelengths.

*Variations* There is a well-known variation of the Phong model, the Blinn-Phong model, which is proposed by Jim Blinn [5]. In the Blinn-Phong model, the term $V \cdot R$ is replaced by $N \cdot H$. Hence, the reflected radiance is

$$I = k_a I_a + k_d I_d \max(N \cdot L, 0) + k_s I_s \left(\max(N \cdot H, 0)\right)^{n'} , \quad (2)$$

where $H = \frac{L+V}{||L+V||}$ is the bisector of $L$ and $V$, which is often called "halfway vector." When $V$ is in the same plane as $N$ and $L$, the angle between $N$ and $H$ is half of that between $V$ and $R$. Hence, it is a close approximation to the original Phong model. This model is illustrated on the right of Fig. 2. The Blinn-Phong model is originally proposed to speed up the computation of the Phong model. It is the default shading model in OpenGL [6] for its efficiency. According to an experimental evaluation [7] with measured reflectance data from real surfaces, the Blinn-Phong model can represent real data more accurately than the original Phong model.

## Application

The Phong reflectance model computes reflected radiance according to the 3D shape, viewing, and lighting configurations. Like many other reflectance models, it is widely used to create computer graphic images. It also provides an analytical tool for radiometric image analysis to understand scene properties such as shape and material from measured image intensities.

## References

1. Phong BT (1975) Illumination for computer generated pictures. Commun ACM 18(6):311–317
2. Cook RL, Torrance KE (1981) A reflectance model for computer graphics. In: SIGGRAPH '81: proceedings of the 8th annual conference on computer graphics and interactive techniques, New York, NY, USA. ACM, pp 307–316
3. Ward GJ (1992) Measuring and modeling anisotropic reflection. In: SIGGRAPH '92: proceedings of the 19th annual conference on computer graphics and interactive techniques, New York, NY, USA, vol 26. ACM, pp 265–272
4. Ashikmin M, Premože S, Shirley P (2000) A microfacet-based brdf generator. In: SIGGRAPH '00: proceedings of the 27th annual conference on computer graphics and interactive techniques, New York, NY, USA. ACM/Addison-Wesley, pp 65–74
5. Blinn JF (1977) Models of light reflection for computer synthesized pictures. In: SIGGRAPH '77: proceedings of the 4th annual conference on computer graphics and interactive techniques, New York, NY, USA. ACM, pp 192–198
6. Shreiner D, Woo M, Neider J, Davis T (2005) OpenGL(R) programming guide: the official guide to learning OpenGL(R), Version 2 (5th Edition) (OpenGL). Addison-Wesley Professional, Boston
7. Ngan A, Durand F, Matusik W (2005) Experimental analysis of brdf models. In: Proceedings of the eurographics symposium on rendering, eurographics association. Association for Computing Machinery, New York, pp 117–226

# Photo-Consistency

Yasutaka Furukawa
Google Inc., Seattle, WA, USA

## Synonyms

Photometric consistency function

## Related Concepts

►Camera Calibration; ►Multi-baseline Stereo

## Definition

Photo-consistency is a scalar function that measures the visual compatibility of a 3D reconstruction with a set of calibrated images.

## Background

Automated 3D reconstruction from images has been a core computer vision problem for years. Multi-view stereo (MVS) is a process of reconstructing 3D structure of an object or a scene from multiple images [1]. MVS assumes calibrated photographs, where camera calibration is often achieved by a calibration chart (e.g., checkerboard patterns) or a Structure from Motion algorithm [2].

In principle, how MVS algorithms recover 3D information is the same as how humans perceive depths with their two eyes, that is, triangulation from correspondences. Therefore, the critical first step of MVS is to establish feature correspondences across multiple input images, where a robust mechanism is necessary to evaluate the goodness of such feature correspondences, which is the role of a photo-consistency function. In a sense, an MVS reconstruction process is to carve out a 2D surface in a 3D space, where photo-consistency scores are high.

## Theory and Examples

Photo-consistency $f(p, V)$ is a scalar function, which measures the visual compatibility of a given 3D reconstruction $p$ with a set of images $V$. Typically, $p$ is a 3D point ($p \in \mathbb{R}^3$), while more sophisticated methods use an oriented point (a 3D point with a surface normal) [3, 4] or a bounded surface region such as a triangle in a polygonal mesh model [5, 6]. For the moment, the visibility information $V$ is assumed to be given for $p$, where details about visibility estimation are referred to a later section.

A simple photo-consistency function at a 3D point $p$ is defined as follows: $p$ is projected into each visible image in $V$, and the similarity of image textures near their projections is computed as photo-consistency. Instead of comparing a single pixel color in each image, a set of pixel colors in each local image region is compared for robustness. More concretely, let $A^i_{uv}$ be the $\tau \times \tau$ rectangular grid of pixel intensities centered at the image projection of $p$ in image $I_i \in V$ (see Fig. 1). Note that $u$ and $v$ are indexes of the rectangular grid, and $\tau = 5, 7$ is typically used. Photo-consistency $f(p, V)$ can be defined as

$$f(p, V) = \sum_{I_i, I_j \in V} \sum_{u,v} (A^i_{uv} - A^j_{uv})^2, \qquad (1)$$

which evaluates a sum of squared differences (SSD) of intensities for every pair of images. SSD score is often used for binocular stereo problems, where a pair of images have a narrow baseline and are acquired under the same or similar lighting conditions. The main reasoning is that the SSD score is sensitive to illumination changes and non-Lambertian effects, which is often the case of MVS problems. Instead of SSD, many MVS algorithms employ the Normalized Cross Correlation measure, which has shown to be more robust:

$$f(p, V) = \sum_{I_i, I_j \in V} \frac{\sum_{u,v} \left(A^i_{uv} - \bar{A}^i_{uv}\right)\left(A^j_{uv} - \bar{A}^j_{uv}\right)}{\sqrt{\left(\sum_{u,v} \left(A^i_{uv} - \bar{A}^i_{uv}\right)^2\right)\left(\sum_{u,v}\left(A^j_{uv} - \bar{A}^j_{uv}\right)^2\right)}}. \qquad (2)$$

$\bar{A}^i_{uv}$ is the average intensity of $\{A^i_{uv}\}$. Another advantage of the NCC score is that the value is guaranteed to be in a range $[-1.0, 1.0]$, where 0.6 or 0.7 is usually a good photo-consistency score, while 0.3 or 0.4 is bad. For a color image, where $A^i_{uv}$ represents a 3D

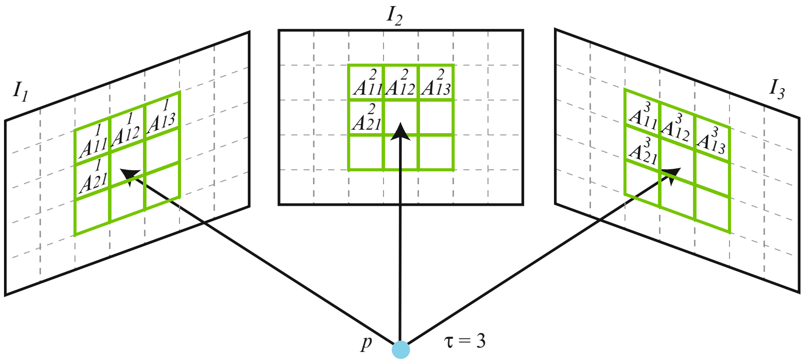

$I_1$
$I_2$
$I_3$
$A^1_{11}$ $A^1_{12}$ $A^1_{13}$
$A^1_{21}$
$A^2_{11}$ $A^2_{12}$ $A^2_{13}$
$A^2_{21}$
$A^3_{11}$ $A^3_{12}$ $A^3_{13}$
$A^3_{21}$
$p$
$\tau = 3$

**Photo-Consistency, Fig. 1** A simple photo-consistency evaluation starts by projecting a 3D point $p$ into each visible image and collecting pixel colors $\{A^i_{uv}\}$ in each local image region

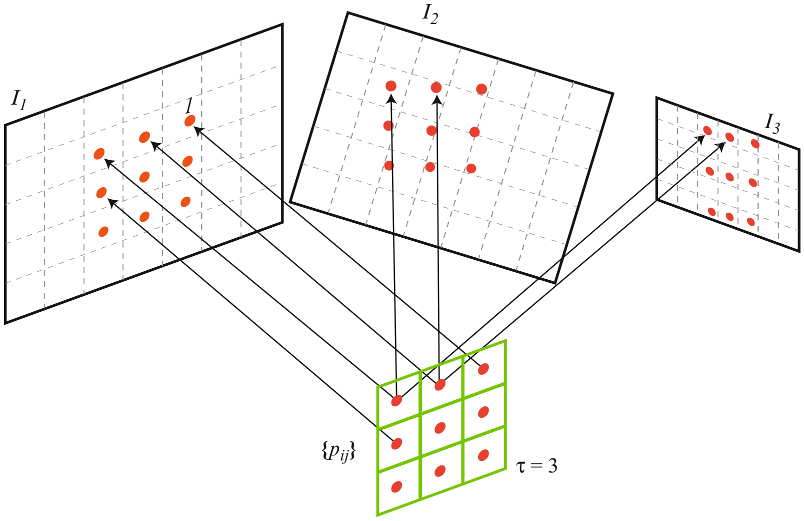


**Photo-Consistency, Fig. 2** More appropriate and accurate photo-consistency evaluation is to sample points in 3D, then project them into each image, which handles differences in camera rotations ($I_2$) and resolutions ($I_3$) properly

vector (red, green, blue), the formula slightly changes its form

$$f(p,V) = \sum_{I_i,I_j\in V} \frac{\sum_{u,v}\left(A^i_{uv}-\bar{A}^i_{uv}\right)^T\cdot\left(A^j_{uv}-\bar{A}^j_{uv}\right)}{\sqrt{\left(\sum_{u,v}\left|A^i_{uv}-\bar{A}^i_{uv}\right|^2\right)\left(\sum_{u,v}\left|A^j_{uv}-\bar{A}^j_{uv}\right|^2\right)}},$$

where the average intensity $\bar{A}^i_{uv}$ is computed for each color channel independently.

This photo-consistency function works well for a simple scene, where images in $V$ have similar resolutions, distances to the point $p$, and rotational parameters. For more complicated scenes, more appropriate evaluation is necessary, which is to first sample a rectangular grid of 3D points $\{p_{ij}\}$ around $p$ in the 3D space, project all the points into each image, and sample pixel colors at their projections (see Fig. 2). Typically, 3D points are sampled on a plane that is front parallel to one of the images in $V$ ($I_i$ in Fig. 2), so that image distances between the adjacent projected points are roughly one pixel. Sampled pixel colors are used in exactly the same way as before (Eq. 1)–(3) to compute photo-consistency scores. Figure 2 illustrates that this photo-consistency function takes into account camera rotation ($I_2$) and resolution ($I_3$) differences to sample colors at the correct pixel locations.

## Advanced Photo-Consistency Functions

For robustness, photo-consistency functions often measure the similarity of image textures over a local region instead of at a single point. In this sense, a natural input to the function should be a 3D point plus some spatial support. One example is an oriented point, which is a combination of a 3D location ($\mathbb{R}^3$) and a surface normal ($\mathbb{S}^2$), which essentially uses a tangent plane approximation of a surface to sample 3D points [3, 4]. Pons and Vu et al. initialize their reconstruction as a polygonal mesh model, and in the process of iterative mesh deformation, photo-consistency is evaluated on the polygonal mesh model directly [5, 6].

### Visibility Estimation

Photo-consistency requires visibility information $V$ as an input. However, it is not easy to obtain good visibility without a 3D model, simply because occlusion information is unknown. Similarly, without visibility, it becomes difficult to obtain accurate 3D reconstructions – chicken-and-egg problem. Furthermore, visibility should reflect certain photometric factors such as specular highlights – images with specular highlights should be excluded from $V$. Fortunately, researchers found out that accurate visibility is not necessary to obtain a successful 3D reconstruction. Instead, a "robust" photo-consistency function can be used to ignore outlier images in $V$, while overestimating $V$. There are several approaches in robustifying photo-consistency functions. One approach is to ignore images whose photo-consistency function scores are worse than a predetermined threshold, which is simple but has proven to work well [3]. Vogiatzis and Hernández et al. proposed more sophisticated approach that handles outliers systematically, which boosted their reconstruction accuracy [7].

## References

1. Seitz SM, Curless B, Diebel J, Scharstein D, Szeliski R (2006) A comparison and evaluation of multi-view stereo reconstruction algorithms. CVPR 1:519–528
2. Hartley R, Zisserman A (2004) Multiple view geometry in computer vision. Cambridge University Press, Cambridge/ New York
3. Furukawa Y, Ponce J (2010) Accurate, dense, and robust multi-view stereopsis. IEEE Trans Pattern Anal Mach Intell 32(8):1362–1376
4. Habbecke M, Kobbelt L (2007) A surface-growing approach to multi-view stereo reconstruction. CVPR
5. Pons JP, Keriven R, Faugeras O (2007) Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. IJCV 72(2):179–193
6. Vu HH, Keriven R, Labatut P, Pons JP (2009) Towards high-resolution large-scale multi-view stereo. In: Conference on computer vision and pattern recognition (CVPR), Miami
7. Vogiatzis G, Esteban CH, Torr PHS, Cipolla R (2007) Multi-view stereo via volumetric graph-cuts and occlusion robust photo-consistency. IEEE Trans Pattern Anal Mach Intell 29(12):2241–2246

# Photogrammetry

Konrad Schindler[1] and Wolfgang Förstner[1,2]
[1] ETH Zürich, Zürich, Switzerland
[2]Universität Bonn, Bonn, Germany

## Definition

Photogrammetry is the science and technology of obtaining information about the physical environment from images, with a focus on applications in surveying, mapping and high-precision metrology. The aim of photogrammetry is to provide automated or semi-automated procedures for these engineering tasks, with an emphasis on a specified accuracy, reliability and completeness of the extracted information.

## Background

Photogrammetry is a long-established engineering discipline, which dates back to the middle of the nineteenth century, shortly after the invention of the photographic process. It has its roots in surveying, predominantly for aerial mapping of the earth's surface, although terrestrial "close-range" photogrammetry has always been an integral part of the discipline. Traditionally photogrammetry has emphasized 3D *geometric* modeling of the environment, since in an interactive setting this implicitly encompassed the semantic interpretation of the image content. The use of geospatial imagery with the primary purpose to infer semantic object properties from radiometric intensities is often referred to as "remote sensing." Today these two fields overlap both in terms of methodology and of applications.

### Methods

The goal of photogrammetry is to extract geometric and semantic information from imagery.

In terms of *geometric* processing methods, the photogrammetric measurement process is essentially an application of structure-from-motion theory [6], mostly (but not exclusively) with calibrated camera intrinsics. In fact a large part of the theory of camera calibration and camera orientation was first developed and applied in photogrammetry, including camera orientation from 2D-to-3D correspondences [1, 5], relative camera orientation from 2D-to-2D correspondences [3], and bundle adjustment [2].

The methods for *semantic* interpretation comprise the entire armamentarium of image understanding, from early rule-based systems [14] through model-based object recognition [18] to statistical learning with modern Bayesian techniques [16]. For a recent overview, see [12]. Due to the complexity of the task, only semi-automatic methods have so far found their way into commercial software and operational production pipelines.

### Relation to Computer Vision

Since the advent of digital images in the 1970s, a main goal has been to automate the photogrammetric process, and photogrammetrists have developed or adopted pattern recognition methods for tasks such as interest point extraction [15], feature matching and dense stereo reconstruction [7, 8], semantic segmentation [11], and object category detection [4]. Thus, the science of photogrammetry is increasingly converging with computer vision and image understanding. Still photogrammetry, being a practical engineering discipline, tends to put greater emphasis on a defined (usually high) accuracy, reliability, and completeness than on total automation.

### Recent Developments

Since the 1990s range images captured directly with airborne and terrestrial laser scanners have gained popularity with both practitioners and researchers in geo-information, and have become a second main data source of photogrammetry [17].

Hand in hand with that development, it has become a standard routine to determine approximate or even final sensor orientations directly, rather than indirectly from observed points. The practice of observing the position and attitude of the camera during flight missions directly with a highly accurate GNSS (global navigation satellite system) receiver and IMU (inertial measurement unit) is known as *direct georeferencing*.

With the advent of high-resolution satellite sensors, spaceborne images from both optical and microwave sensors nowadays also serve as input data for the photogrammetric process. Mobile mapping systems mounted on vehicles have lead to a growing interest in large-scale photogrammetric mapping from the ground.

## Application

The most important application field of photogrammetry is topographic mapping of the earth's surface at different scales. The overwhelming majority of all existing maps have been created through photogrammetric processing of airborne or spaceborne imagery. However, photogrammetry is also prominent for small-scale mapping down to single villages, mines, etc. A related endeavor has been the mapping of other planets in the solar system from images taken by spacecraft.

Non-topographic applications for a long time occupied only a small fraction of the market. They used to be subsumed under the term "close-range photogrammetry," the main application fields being industrial metrology (e.g., aircraft, ships, vehicle parts), construction, cultural heritage documentation, forensics, and the medical domain.

Although mapping remains the dominant application area, the boundary between photogrammetry and 3D computer vision is dissolving more and more. Today tasks like visual driver assistance and robot navigation, motion capture, virtual and augmented reality, object tracking, etc. are by many also considered applications of photogrammetry.

Technology transfer between academia and industry has always been well established in photogrammetry, via the International Society for Photogrammetry and Remote Sensing (ISPRS, http://www.isprs.org) and the long-running biennial Photogrammetric Week (http://www.ifp.uni-stuttgart.de/publications/phowo.html).

## References

1. Abdel-Aziz YI, Karara HM (1971) Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. Proceedings of the Symposium on Close-Range Photogrammetry, Falls Church, VA, pp 1–18

2. Brown DC (1958) A solution to the general problem of multiple station analytical stereotriangulation. RCA-MTP Data Reduction Technical Report No. 43, Patrick Airforce Base, FL
3. Finsterwalder S (1897) Die geometrischen Grundlagen der Photogrammetrie. Jahresbericht der Deutschen Mathematiker-Vereinigung 6(2):1–41
4. Grabner H, Nguyen TT, Gruber B, Bischof H (2007) Online boosting-based car detection from aerial images. ISPRS J Photogramm Remote Sens 63(3):382–396
5. Grunert JA (1841) Das Pothenotische Problem in erweiterter Gestalt nebst Über seine Anwendungen in der Geodäsie. Grunerts Archiv für Mathematik und Physik 1: 238–248
6. Hartley RI, Zisserman A (2004) Multiple view geometry in computer vision, 2nd edn. Cambridge University Press, Cambridge/New York
7. Helava UV (1978) Digital correlation on photogrammetric instruments. Photogrammetria 34(1):19-41
8. Kelly RE, McConnell PRH, Mildenberger SJ (1977) The Gestalt photomapping system. Photogramm Eng Remote Sens 42(11):1407–1417
9. Kraus K (2007) Photogrammetry: geometry from images and laser scans, 2nd edn. Walter de Gruyter, Berlin/ New York
10. Luhmann T, Robson S, Kyle S, Harley I (2006) Close Range Photogrammetry: Principles, Techniques and Applications. Whittles, Dunbeath
11. Mayer H, Laptev I, Baumgartner A (1998) Multi-scale and snakes for automatic road extraction. Proceedings of the 5th European Conference on Computer Vision (ECCV), Freiburg, Germany, pp 720–733
12. Mayer H (2008) Object extraction in photogrammetric computer vision. ISPRS J Photogramm Remote Sens 63(2): 213–222
13. McGlone JC (ed.) (2013) Manual of Photogrammetry. 6th edn. American Society for Photogrammetry and Remote Sensing, Virginia
14. McKeown DM, Harvey WA, McDermott J (1985) Rule-Based interpretation of aerial imagery. IEEE Trans Pattern Anal Mach Intell 7(5):570–585
15. Paderes FC, Mikhail EM, Förstner W (1984) Rectification of single and multiple frames of satellite scanner imagery using points and edges as control. NASA Symposium in Mathematical Pattern Recognition and Image Analysis. Houston, TX, pp 309–400
16. Stoica R, Descombes X, Zerubia J (2004) A Gibbs point process for road extraction in remotely sensed images. Int J Comput Vis 57(2):121–136
17. Vosselman G, Mass H-G (eds) (2010) Airborne and terrestrial laser scanning. Whittles, Dunbeath
18. Weidner U, Förstner W (1995) Towards automatic building extraction from high-resolution digital elevation models. ISPRS J Photogramm Remote Sens 50(4): 38–49

# Photometric Consistency Function

▶Photo-Consistency

# Photometric Invariants

Todd Zickler
School of Engineering and Applied Science, Harvard University, Cambridge, MA, USA

## Related Concepts

▶Image Decompositions

## Definition

A photometric invariant is a function of an image, or a function of a set of images, that is discriminative with respect to some scene properties and independent of others.

## Background

J. J. Gibson was the first to write extensively about computing useful "invariants" from visual observations. According to Gibson, an invariant represents the extraction of persistent scene properties despite changing environmental conditions and therefore enables the concurrent awareness of both persistence and change within a scene. While he does not use the term "photometric invariant," he describes the concept quite well [10]:

> . . . the illumination can change in amount, in direction, and in spectral composition. Some features of any optic array in the medium will change accordingly. There must be invariants for perceiving surfaces, their relative layout, and their relative reflectances. They are not yet known, but they almost certainly involve ratios of intensity and color among parts of the array.

The notion of a photometric invariant is different from, but related to, the mathematical definition of an invariant. In mathematics, an invariant is a property that remains unchanged when certain transformations are applied, whereas a photometric invariant is a function of an image (or set of images) that persists despite changes in certain scene properties and conditions.

Photometric invariants are also related to, but distinct from, image decompositions based on the explicit separation of an image according to scene characteristics, such as shading and lightness [11], diffuse and specular reflectance [12], or illuminant and spectral reflectance [1]. Computing such explicit decompositions is ill-posed, whereas photometric invariants can be computed directly from image intensities, often in closed form. In fact, it is not uncommon for photometric invariants to be used as initialization for iterative image decomposition techniques.

Photometric invariants are used to discriminate among the scene properties of interest while eliminating dependence on other "distractors." Insensitivity and discriminability are usually in conflict, and in general one cannot obtain one without sacrificing some of the other. For example, it can be shown that there is no photometric invariant computed from grayscale imagery that can exactly discriminate between surface shapes under variable lighting, because given any two images, one can always find a single surface to explain them [2]. A high-performing invariant, therefore, is one that provides a *balance* between insensitivity and discriminability and does so in a way that is appropriate for the desired visual task and operating environment. For this reason, performance of photometric invariants should be conducted empirically, preferably using imagery that is representative of the end goal.

## Theory

An early discovery of a photometric invariant was made by Koenderink and van Doorn [13], who considered matte (Lambertian) surfaces under directional lighting. Under these conditions, certain stationary points of image isophotes cling to parabolic surface points and therefore provide information about surface shape that does not depend on the locations of light sources.

Since then, the community has discovered a diverse collection of photometric invariants, usually written as simple functions of intensity and color at the image projection of one or more scene points. Each of these invariants is derived by assuming models for lighting, reflectance, and spectral sensors and by doing so in the context of an imaging model that expresses the measurements in the $k$th channel as

$$I_k(x, L) = \int_{\Lambda} \int_{\mathbb{S}^2} c_k(\lambda) L(\lambda, \omega) f(x, \lambda, n, \omega, \omega') \max(0, \langle \omega, n \rangle) d\omega d\lambda. \quad (1)$$

Here, $\{c_k(\lambda)\}_{k=1...K}$ are the spectral responses of a camera with $K$ channels; $f(x, \lambda, n, \omega, \omega')$ is the spatially varying, spectral bidirectional reflectance distribution function (BRDF) at the back projection of image point $x$ evaluated with surface normal $n \in \mathbb{S}^2$, light direction $\omega \in \mathbb{S}^2$, and view direction $\omega' \in \mathbb{S}^2$; and $L(\lambda, \omega)$ is the (spatially uniform) spectral radiance distribution that illuminates the scene.

The photometric invariants differ in their assumptions about the sensors $\{c_k(\lambda\}$ as well as their assumptions regarding reflectance and lighting. In all cases, reflectance and lighting models are based on factored representations that separate the variation with respect to spatial position, wavelength, and angular geometry:

$$f(x, \lambda, \theta) = \sum_{n=1}^{N} m_n(x, \lambda) g_n(\theta), \quad (2)$$

with $\theta \triangleq \{n, \omega, \omega'\}$ defined to simplify notation and

$$L(\lambda, \omega) = \sum_{m=1}^{M} e_m(\lambda) \ell_m(\omega). \quad (3)$$

The assumptions that underlie each derivation provide some intuition for when each invariant might be usefully employed. These assumptions are categorized below; the resulting invariants are listed in Tables 1 and 2; and visualizations for some of them appear in Fig. 1. Note that empirical evaluations of many of these invariants reveal them to be useful when the underlying assumptions are only approximately satisfied.

**Reflectance model.** Common factored models are the Lambertian model ($N = 1$, $g_1(\theta) =$ constant in Eq. 2) and the dichromatic model with a neutral interface ($N = 2$, $g_1(\theta) =$ constant, $m_2(x, \lambda) = m_2(x)$). Invariants for both of these cases are listed separated in Tables 1 and 2.

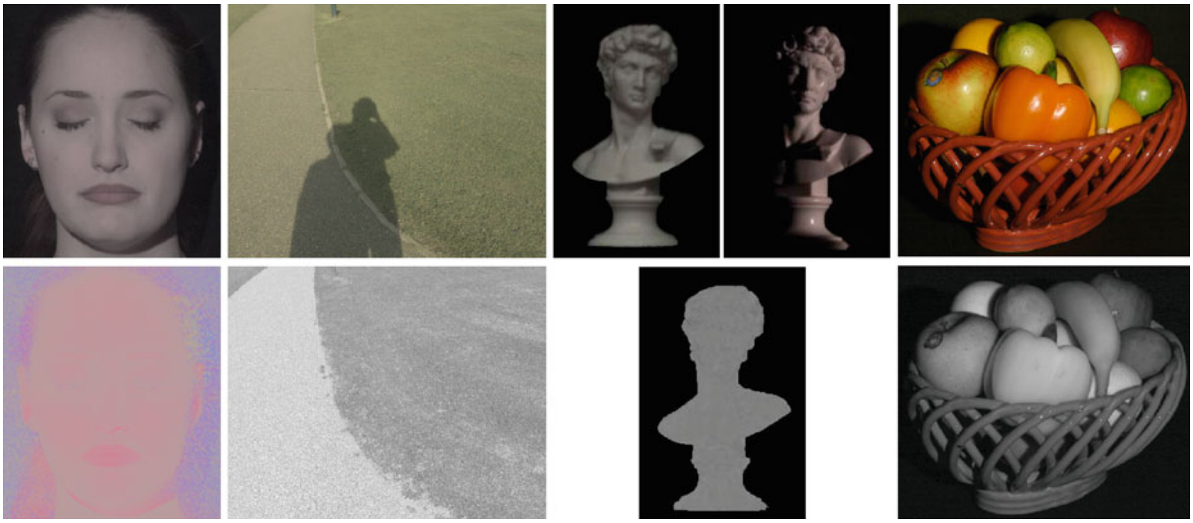
**Lighting model.** The most common instance of the factored lighting model has the same spectrum in every direction ($M = 1$ in Eq. 3), with the associated spectrum $e_1(\lambda)$ being either arbitrary, "even" ($e_1(\lambda) =$ constant, so that $L(\lambda, \omega) = \ell(\omega)$), or Planckian and thus completely defined by its color temperature $T$: $e_1(\lambda; T)$.

**Photometric Invariants, Table 1** Photometric invariants for surface reflectance of the form $f(x, \lambda, \theta) = m(x, \lambda) g(\theta)$, which includes the Lambertian model as a special case. Each is independent of certain scene properties and derived from assumed models of lighting and sensors

| Expression | Independent of | Comments |
|---|---|---|
| $\frac{I_1(x)}{I_1(x) + I_2(x) + I_3(x)}$ | Geometry $\theta$<br>Intensity $\ell(\omega)$ | – “Normalized RGB”<br>– Related: chromaticity, hue, saturation<br>– Reference: *e.g.*, [9]<br>– Lighting: $L(\omega, \lambda) = \ell(\omega) e(\lambda)$<br>– Sensors: general |
| $\frac{I(x_1)}{I(x_2)}$ | Geometry $\theta$<br>Intensity $\ell(\omega)$<br>Spectrum $e(\lambda)$ | – “Reflectance ratio”<br>– $x_1,\ x_2$ must have same surface normal<br>– References: [7, 15]<br>– Lighting: $L(\omega, \lambda) = \ell(\omega) e(\lambda)$<br>– Sensors: von Kries |
| $\frac{I_1(x_1) I_2(x_2)}{I_1(x_2) I_2(x_1)}$ | Geometry $\theta$<br>Intensity $\ell(\omega)$<br>Spectrum $e(\lambda)$ | – Reference: [9]<br>– Lighting: $L(\omega, \lambda) = \ell(\omega) e(\lambda)$<br>– Sensors: von Kries |
| $\alpha \log \frac{I_1}{I_3} + \beta \log \frac{I_2}{I_3}$ | Geometry $\theta$<br>Intensity $\ell(\omega)$<br>Spectrum $e(\lambda)$ | – $(\alpha_1, \alpha_2)$ depend on camera sensors<br>– Reference: [5]<br>– Lighting: $L(\omega, \lambda) = \ell(\omega) e(\lambda)$ with $e(\lambda)$ Planckian<br>– Sensors: narrow band |
| $\left( \frac{\partial I_2}{\partial x} I_1 - I_2 \frac{\partial I_1}{\partial x} \right) / I_1^2$ | Geometry $\theta$<br>Intensity $\ell(\omega)$<br>Spectrum $e(\lambda)$ | – Reference: [8]<br>– Sensors: $c_1(\lambda)$ and $c_2(\lambda)$ must approximate Gaussian spectral derivatives |
| $\frac{I(x, L_1)}{I(x, L_2)}$ | Material $m(x, \lambda)$ | – “Photometric ratio”<br>– Reference: [17]<br>– Lighting: $L_j(\omega, \lambda) = \ell_j(\omega) e(\lambda)$<br>– Sensors: general |

**Photometric Invariants, Table 2** Photometric invariants for surface reflectance of the form $f(x, \lambda, \theta) = m_1(x, \lambda) g_1(\theta) + m_2(x) g_2(\theta)$, which includes the dichromatic model with neutral interface as a special case. Each is independent of certain scene properties and derived from assumed models of lighting and sensors

| Expression | Independent of | Comments |
|---|---|---|
| $\operatorname{atan} \left( \frac{\sqrt{3}(I_2 - I_3)}{(2I_1 - I_2 - I_3)} \right)$ | Geometry $\theta$<br>Intensity $\ell(\omega)$ | – “Hue”<br>– Reference: [4]<br>– Lighting: $L(\omega, \lambda) = \ell(\omega)$<br>– Sensors: general |
| $\alpha_1 I_1 + \alpha_2 I_2 + \alpha_3 I_3$ | Intensity $\ell(\omega)$<br>Component $m_2(x) g_2(\theta)$ | – “Color subspace”<br>– $\{\alpha_i\}$ depend on $e(\lambda)$, sensors<br>– Reference: [18]<br>– Lighting: $L(\omega, \lambda) = \ell(\omega) e(\lambda)$; generalizes to $M > 1$ in Eq. 3<br>– Sensors: general |
| $\frac{I_1(L_1) I_2(L_2) - I_2(L_1) I_1(L_2)}{I_1(L_1) I_3(L_2) - I_3(L_1) I_1(L_2)}$ | Geometry $\theta$ | – “Ratio of determinants”<br>– Generalizes to BRDF with $N > 2$ in Eq. 2<br>– Reference: [14]<br>– Lighting: $L_j(\omega, \lambda) = \ell_j(\omega) e(\lambda)$<br>– Sensors: general |



**Photometric Invariants, Fig. 1** Visualization of photometric invariants (*bottom row*) computed from one or two input images (*top row*). From left to right that based on normalized RGB, Planckian lighting [5], ratio of determinants [14], and color subspaces [18]

**Color or grayscale.** Invariants can be computed from multiple spectral measurements (usually three), which are denoted by $\{I_k(x)\}_{k=1,2,3}$ in Tables 1 and 2. In some cases, they are computed from a single spectral measurement (a "grayscale" image), which is denoted by $I(x)$.

**Sensor model.** For invariants based on color, the three sensors $\{c_k(\lambda)\}_{k=1,2,3}$ may be arbitrary, or they may be delta functions (*i.e.*, "narrow-band" sensors): $c_k(\lambda) = \delta(\lambda - \lambda_k)$. In between these two extremes are possibly overlapping sensors that nonetheless support spectral relighting by independent per-channel gain factors. Such sensors are said to support *von Kries adaptation*, and, as described in [3], they must satisfy a tensor rank constraint when integrated against all pairs of material spectral $(m_n(\cdot, \lambda))$ and lighting spectra $(e_m(\lambda))$ that could possibly exist in the operating environment.

**Single point or multiple points.** Invariants can be computed independently at each pixel, or they may be computed by combining measurements at distinct image points (denoted by $I(x_i)$ in the tables).

**Single image or multiple images.** Similarly, invariants can be computed from measurements in a single image or by combining measurements from multiple images captured under distinct lighting environments (denoted by $I(x, L_j)$ in the tables).

## Application

Photometric invariants can be viewed as an alternative to learning-based approaches that attempt to model the appearance of persistent scene properties (shape, reflectance patterns, *etc.*) over all configurations of lighting, viewpoint, and other distractors. Instead of modeling this appearance variation, they "hard-code" an invariance, perhaps at the expense of discriminability. When applicable, the main advantages of invariant-based approaches are their computational efficiency and reduced requirement for training data. Whether a learning-based approach or an invariant-based approach (or a combination of the two) is more desirable depends on the visual task and the operating environment.

Multiple-point photometric invariants that are independent of geometry $\theta$ and light intensity $\ell(\omega)$ isolate surface reflectance information, and since they do not depend on light and viewing conditions, they may improve performance on object recognition and image indexing tasks [7, 15]. In some cases, performance may be further improved by using an invariant that is also independent of illuminant spectrum [9].

Invariants that are computed at a single point and are independent of geometry can also improve material-based segmentation and boundary detection by reducing the occurrence of false boundaries due to shading and specular highlights. In some cases, this can be achieved from a single image [8, 18], and additional images can be used to handle more complex BRDFs [14].

The Planckian invariant [5] has the unique property of being independent of illuminant spectrum while also being computed at a single point. For this reason, it can be used for detecting and removing shadows in images that contain mixtures of distinct illuminant spectra [6].

Photometric invariants that are computed at a single point and are independent of material properties can be used to extract surface shape information. For diffuse surfaces, photometric ratios [17] can provide access to surface curvature information independent of surface albedo, and for surfaces described by the dichromatic model, color subspaces [18] can isolate the diffuse component and therefore improve the performance of shape-from-shading, photometric stereo, and a variety of other Lambertian-based vision algorithms.

Finally, as mentioned above, photometric invariants may be used as initialization for explicit image decompositions. The color subspace invariant [18] (and one closely related to it [16]) can be useful when decomposing an image into its diffuse and specular components, and the reflectance ratio [7, 15] is closely related to retinex-like algorithms for decomposing an image into shading and lightness [11].

## References

1. Brainard D, Freeman W (1997) Bayesian color constancy. J Opt Soc Am A 14(7):1393–1411
2. Chen H, Belhumeur P, Jacobs D (2002) In search of illumination invariants. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)
3. Chong H, Gortler S, Zickler T (2007) The von Kries hypothesis and a basis for color constancy. In: Proceedings of the IEEE international conference on computer vision, Rio de Janeiro
4. D'Zmura M, Lennie P (1986) Mechanisms of color constancy. J Opt Soc Am A 3(10):1662–1672
5. Finlayson G, Hordley S (2001) Color constancy at a pixel. J Opt Soc Am A 18(2):253–264
6. Finlayson G, Hordley S, Drew M (2006) Removing shadows from images. In: Proceedings of the European conference on computer vision (ECCV), Graz
7. Funt B, Finlayson G (1995) Color constant color indexing. IEEE Trans Pattern Anal Mach Intell 17(5):522–529
8. Geusebroek J, van den Boomgaard R, Smeulders A, Geerts H (2001) Color invariance. IEEE Trans Pattern Anal Mach Intell, 23(12):1338–1350
9. Gevers T, Smeulders A (1997) Color based object recognition. In: Image analysis and processing. Springer, Berlin/New York, pp 319–326
10. Gibson J (1979) The ecological approach to visual perception. Houghton Mifflin, Boston
11. Horn B (1974) Determining lightness from an image. Comput Graph Image Process 3(4):277–299
12. Klinker G, Shafer S, Kanade T (1988) The measurement of highlights in color images. Int J Comput Vis 2(1):7–32
13. Koenderink J, van Doorn A (1980) Photometric invariants related to solid shape. Optica Acta 27:981–996
14. Narasimhan S, Ramesh V, Nayar S (2003) A class of photometric invariants: Separating material from shape and illumination. In: Proceedings of the IEEE international conference on computer vision, Nice
15. Nayar S, Bolle R (1996) Reflectance based object recognition. Int J Comput Vis 17(3):219–240
16. Tan R, Ikeuchi K (2005) Separating reflection components of textured surfaces using a single image. IEEE Trans Pattern Anal Mach Intell 27(2):178–193
17. Wolff L, Fan J (1994) Segmentation of surface curvature with a photometric invariant. J Opt Soc Am A 11(11): 3090–3100
18. Zickler T, Mallick S, Kriegman D, Belhumeur P (2008) Color subspaces as photometric invariants. Int J Comput Vis 79(1):13–30

# Photometric Stereo

Ronen Basri
Department of Computer Science And Applied
Mathematics, Weizmann Institute of Science,
Rehovot, Israel

## Related Concepts

►Active Stereo Vision; ►Lambertian Reflectance

## Definition

Photometric stereo is the problem of recovering the 3-dimensional shape of a stationary scene given a collection of images of the scene taken under variable lighting conditions.

## Background

The amount of light reflected by an object is a function of the incoming light, the shape of the object, and its material properties. As a function of shape, the light reflected by each surface point depends on the inclination of the surface (the surface normal) at that point relative to the light sources. Early work in photometric stereo [1] assumes that the lighting conditions are given and that the material properties of the object are known. A collection of images of an object taken under varying lighting conditions (with both the object and camera stationary) can then be used to compute the normals to the surface of the object and subsequently its 3-dimensional shape. More recent photometric stereo methods can additionally recover the lighting conditions and material properties as part of the process.

Photometric stereo (PS) can be contrasted with shape from shading (SFS), the problem of recovering the shape of an object from a *single* image [2]. The problem of SFS is generally ill posed, and its solution typically requires complete knowledge of lighting and material properties. In addition, SFS is often cast as a partial differential equation (PDE), and so its solution relies on boundary conditions. These boundary conditions generally require knowledge of the 3D coordinates of a subset of the points on the sought shape. In contrast, by using a collection of images, PS generally leads to more robust solutions; it can often be solved also in the absence of knowledge of lighting conditions or material properties, requires no boundary conditions, and its solution is generally algebraic.

## Theory

Let $I_1(x,y), I_2(x,y), \dots I_k(x,y)$, $(x,y) \in \Omega \subset \mathbb{R}^2$, be a collection of $k$ images depicting a stationary scene pictured by a stationary camera. Suppose further that each image $I_i$ is taken under different lighting, which is denoted by $L_i$. The objective of photometric stereo is to recover the 3D shape of the scene, represented by $z(x,y)$, which depicts the depth value $z$ at each point $(x,y)$.

In general terms, let $\hat{\mathbf{n}}(x,y) \in \mathcal{S}^2$ denote the normal to the surface at $(x,y,z(x,y))$ ($\|\hat{\mathbf{n}}(x,y)\| = 1$), and let $\rho(x,y)$ denote the reflectance properties of each scene point. The light reflected by a scene point $(x,y,z(x,y))$ is determined by the surface normal $\hat{\mathbf{n}}(x,y)$, the reflectance properties $\rho(x,y)$, and the lighting conditions $L_i$ and can thus be expressed as a function $R(\hat{\mathbf{n}}, \rho, L_i)$. (Note that this expression does not model the effects of cast shadows or interreflections, as these effects depend more globally on the shape of the observed object.) Each image then provides a constraint on the surface normal of the form $I_i(x,y) = R(\hat{\mathbf{n}}, \rho, L_i)$. With sufficiently many images these constraints can be used to recover the surface normal at each point, $\hat{\mathbf{n}}$, and, subsequently, the surface $z(x,y)$.

For a concrete example, due to the pioneering work of Woodham [1], consider a scene composed of a matte surface whose reflectance is described by the Lambertian model, and suppose that in each image the scene is illuminated by a single directional source. Specifically, a *directional* source (also referred to as a *point source at infinity*) is expressed by a vector $\mathbf{l}_i \in \mathbb{R}^3$ in the direction of the source whose magnitude represents the light intensity. According to the Lambertian law, the light reflected by a point $(x,y,z(x,y))$ with normal $\hat{\mathbf{n}}(x,y)$ and albedo $\rho(x,y)$ (the *albedo* of a scene point is a material property representing the fraction of incident light that is reflected by the surface at that point) is given by

$$I_i(x,y) = \rho(x,y)\mathbf{l}_i^T \hat{\mathbf{n}}(x,y), \qquad (1)$$

where the superscript $T$ denotes the transpose operator. This law is applicable as long as $\mathbf{l}_i^T \hat{\mathbf{n}}(x,y) \geq 0$; otherwise, the point is in shadow (commonly referred to as *attached shadow*). Note that for this equation, it is assumed that the camera is calibrated so that the amount of light recorded by the camera is identical to the amount of light reflected by the surface.

Suppose now that $k$ such images $I_1, \dots, I_k$ are obtained, in which the scene is illuminated by directional light sources $\mathbf{l}_1, \dots \mathbf{l}_k$, respectively. Let $\mathbf{I}_i \in \mathbb{R}^p$ be a vector arrangement of the pixel intensities in $I_i(x,y)$, $\mathbf{I}_i = (I_i(x_1,y_1), \dots, I_i(x_p,y_p))^T$, where $p$ denotes the number of discrete pixels in $\Omega$. Let $M = [I_1, \dots I_k]^T$ be a $k \times p$ matrix whose rows include the intensity measurements in all images. Let $L = [\mathbf{l}_1, \dots, \mathbf{l}_k]^T$ be a $k \times 3$ matrix containing all $k$ light sources. Finally, let $S = [\rho(x_1,y_1)\hat{\mathbf{n}}(x_1,y_1), \dots, \rho(x_p,y_p)\hat{\mathbf{n}}(x_p,y_p)]$ be a $3 \times p$ matrix whose columns contain the surface normals of the scene points scaled by the corresponding albedo (with the columns of $M$ and $S$ organized in correspondence). $M$ is referred to as the measurement matrix, $L$ as the lighting matrix, and $S$ as the shape matrix. Then the Lambertian law for all the input images can be summarized by the following matrix equation:

$$M = LS. \tag{2}$$

Assuming that all the lighting directions and intensities are known (so $L$ is known) and that $L$ includes three linearly independent rows, then $S$ can be determined by solving the linear, possibly overdetermined equation system above. In particular, if $k = 3$ and $L$ is invertible, then

$$S = L^{-1}M. \tag{3}$$

Once $S$ is recovered the surface normals of the scene can be recovered by normalizing each column of $S$, i.e., let $\mathbf{s}^i$ denotes the $i$'th column of $S$ then

$$\hat{\mathbf{n}}(x_i, y_i) = \frac{\mathbf{s}^i}{\|\mathbf{s}^i\|} \tag{4}$$

$$\rho(x_i, y_i) = \|\mathbf{s}^i\|. \tag{5}$$

Note that the collection of surface normals of a differentiable shape uniquely determines the depth values of its bounding surface up to an additive constant.

## Integrability

To further obtain an explicit recovery of the depth values $z(x, y)$, *integrability* (or *consistency*) of the recovered normals can be imposed. For integrability the surface $z(x, y)$ is assumed to be differentiable, and the normals can be expressed in terms of its derivatives. Under these conditions, it can be readily shown that the normal at a point $(x, y, z(x, y))$ is given by

$$\hat{\mathbf{n}}(x, y) = \frac{(z_x, z_y, -1)}{\sqrt{z_x^2 + z_y^2 + 1}}, \tag{6}$$

where $z_x$ and $z_y$ represent the two partial derivatives of $z$, $z_x = \partial z/\partial x$ and $z_y = \partial z/\partial y$. An estimate of these partial derivatives can be obtained from the recovered shape matrix $S$ using the ratios $z_x = -s_1/s_3$ and $z_y = -s_2/s_3$, where $s = (s_1, s_2, s_3)^T$ is the column of $S$ corresponding to point $(x, y)$. Finally, a forward discretization of the partial derivatives is used to obtain the following first-order difference equations

$$z(x + h, y) - z(x, y) = hz_x(x, y) \tag{7}$$

$$z(x, y + h) - z(x, y) = hz_y(x, y), \tag{8}$$

where $h$ is the (known) meshsize (often $h = 1$). This is an overdetermined system of linear equations in $z(x, y)$ and can be solved to least squares up to an additive constant which can be determined, e.g., by setting $z(x_0, y_0) = 0$ for some point $(x_0, y_0)$.

## Simultaneous Recovery of Shape and Light

The method above assumes that the intensity and direction of the light sources are known and uses this information to recover the shape and albedo of the object. However, these derivations can still be used even when the lighting conditions are unknown. The main observation is that in the absence of noise, the measurement matrix $M$ has rank 3, and so it can be factored to recover the lighting and shape up to a certain ambiguity [3]. In addition, when rank$(M)$ exceeds 3, replacing it by its rank 3 approximation can eliminate at least some of the noise in the image.

To see that rank$(M)$ is (at most) 3, note that it is a product of the $k \times 3$ lighting matrix $L$ with the $3 \times p$ shape matrix $S$ (Eq. (2)). Hence, $M$ can be factored using singular value decomposition (SVD) as follows. let $\hat{M}$ be the best rank 3 approximation to $M$, i.e., $\hat{M} = \operatorname{argmin}_{\tilde{M}} \|\tilde{M} - M\|_F$, where $\|.\|_F$ denotes the Frobenius Norm of a matrix. $\hat{M}$ can be readily computed using the SVD decomposition of $M$. Next, let $\hat{M} = U\Sigma V^T$ be the SVD decomposition of $\hat{M}$ and define $\hat{L} = U\sqrt{\Sigma}$ and $\hat{S} = \sqrt{\Sigma}V^T$. Evidently, this decomposition is nonunique, as any $3 \times 3$ nonsingular matrix $A$ can be used to obtain another valid decomposition, i.e.,

$$\hat{M} = (\hat{L}A^{-1})(A\hat{S}). \tag{9}$$

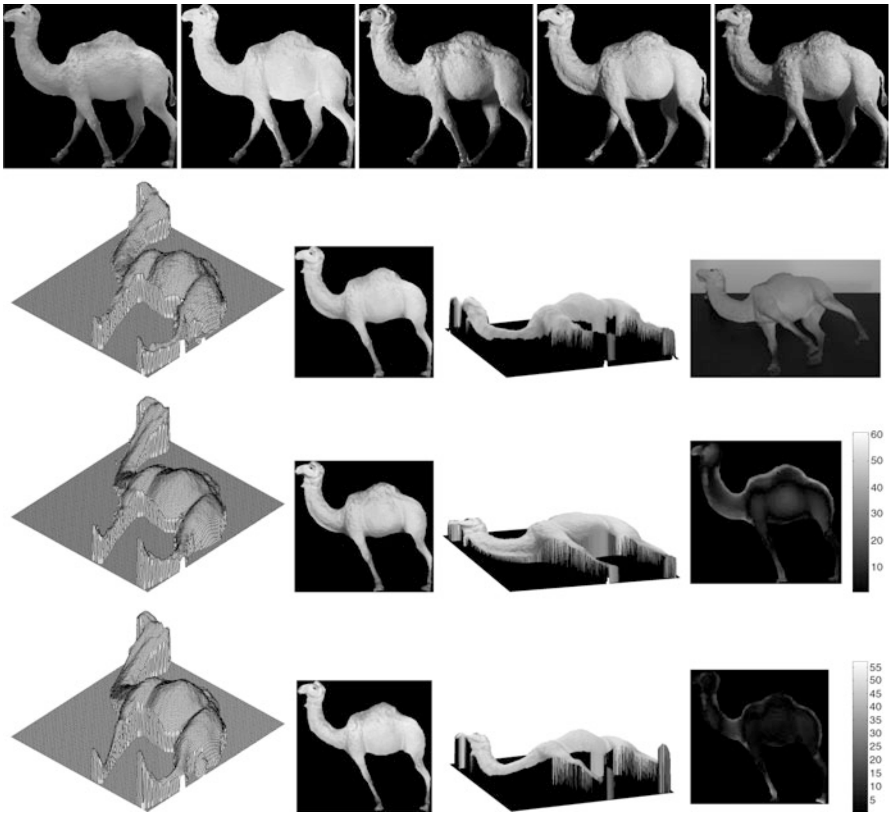
Equation (9) therefore defines a 9-parameter ambiguity – the entries of $A$. This ambiguity can be reduced by imposing integrability [4], as is explained below.

To impose integrability the following equation can be used:

$$z_{xy} = z_{yx}, \tag{10}$$

which holds when $z$ is twice differentiable, $z_{xy} = \partial^2 z/(\partial x \partial y)$ and $z_{yx} = \partial^2 z/(\partial y \partial x)$. Let $A$ denote the ambiguity matrix so that $S = A\hat{S}$. Then

$$z_x = -\frac{s_1}{s_3} = -\frac{\mathbf{a}_1^T \hat{\mathbf{s}}}{\mathbf{a}_3^T \hat{\mathbf{s}}} \tag{11}$$



**Photometric Stereo, Fig. 1** *Top row*: five (of 13) images used for reconstruction. *Second row*: the shape of the camel obtained with a laser scanner (from *left* to *right*: surface, albedo, and albedo-painted surface) and an image taken from roughly the same view (*right*). *Third row*: reconstruction using the first-order method (including shape, albedo, albedo-painted shape, and difference from the laser-scanned surface). *Bottom row*: reconstruction using the second-order method (From [5, 6], used with permission)

$$z_y = -\frac{s_2}{s_3} = -\frac{\mathbf{a}_2^T\hat{\mathbf{s}}}{\mathbf{a}_3^T\hat{\mathbf{s}}}, \qquad (12)$$

where $(s_1, s_2, s_3)^T$ is a column in the unknown shape matrix $S$, $\hat{\mathbf{s}}$ is the corresponding column in the recovered matrix $\hat{S}$, and $\mathbf{a}_i^T$ denotes the $i$th rows of $A$ $(1 \le i \le 3)$. Plugging these into (10),

$$\frac{\partial}{\partial y}\left(\frac{\mathbf{a}_1^T\hat{s}}{\mathbf{a}_3^T\hat{s}}\right) = \frac{\partial}{\partial x}\left(\frac{\mathbf{a}_2^T\hat{s}}{\mathbf{a}_3^T\hat{s}}\right), \qquad (13)$$

a linear equation in the components of $A^T A$ is obtained for every column in $\tilde{S}$:

$$\mathbf{a}_1^T\hat{s}_y\,\mathbf{a}_3^T\hat{s} - \mathbf{a}_1^T\hat{s}\,\mathbf{a}_3^T\hat{s}_y = \mathbf{a}_2^T\hat{s}_x\,\mathbf{a}_3^T\hat{s} - \mathbf{a}_2^T\hat{s}\,\mathbf{a}_3^T\hat{s}_x, \quad (14)$$

with $\hat{s}_x = \partial\hat{s}/\partial x$ and $\hat{s}_y = \partial\hat{s}/\partial y$. This (generally overdetermined) system of equations determines $A$ up to a *generalized* bas-relief (GBR) transformation [4], i.e., a depth estimate $\hat{z}(x, y)$ is obtained that is related to the original depth $z(x, y)$ by

$$\hat{z}(x, y) = \alpha x + \beta y + \gamma z(x, y), \qquad (15)$$

with arbitrary constants $\alpha$, $\beta$, and $\gamma$.

### Photometric Stereo with General Lighting

The derivations above assume that each of the input images is produced by illuminating the object by a single directional source. More recently, Basri et al. [5, 6] proposed an approach to extend this to handle Lambertian objects illuminated by arbitrary combinations of directional and extended light sources. Their algorithm is based on the observation that the light reflected by Lambertian objects, as a function of the surface normal, is a low-pass filtered version of the surrounding ambient light [7–9]. This allows one to describe the images of Lambertian objects, with great accuracy, as linear combinations of a small set (typically 4 or 9) of basis images, which are determined by the low-order spherical harmonic functions of the surface normal. This linear representation can now be exploited to construct factorization algorithms to photometric stereo under general lighting. The experimental results below show an example of a reconstruction achieved with this approach.

## Application

Three-dimensional reconstruction is one of the fundamental tasks of computer vision. Photometric stereo is a reliable method for reconstruction. It is however used mostly in *laboratory conditions* that are partly controlled. This is because its required input should include a stationary scene under variable lighting conditions.

## Open Problems

The majority of existing photometric stereo methods are designed to handle Lambertian objects. Initial work has been devoted to handling objects that exhibit specular reflectance, either simply by removing highlighted pixels [10] or by a detailed modeling of reflectance [11, 12]. Another challenge is to model the effects of cast shadows and interreflections (see, e.g., [13]). Finally, a challenging problem is to recover the 3D shapes of objects when the objects are moving with respect to a light source. Preliminary work in this subject can be found in [14–17].

## Experimental Results

Figure 1 shows a photometric stereo reconstruction obtained with the algorithms proposed in [5, 6] and a comparison to a laser scan of the same object. In this experiment, 13 images of a camel-shaped doll were obtained under varying lighting conditions. The lighting setting was general and involved a number of sources along with reflections from surrounding objects. A factorization method was applied to recover both the lighting and the 3-dimensional shape of the doll. Two results are shown. In the first case reflectance was approximated by a first-order harmonic approximation. This approximation is analogous to assuming that the lighting setting includes a single directional source in addition to a uniform, ambient source. In the second case reflectance was approximated by the more accurate, second-order harmonic approximation. The figure shows a subset of the input images and the laser scanned reconstruction along with the shapes and albedos produced by the photometric stereo algorithms. It can be seen that both methods managed to recover the shape of the camel correctly (the results produced by the second-order method are slightly more accurate), as can be judged by comparing the reconstructions to the shape produced by the laser scanner.

## References

1. Woodham RJ (1980) Photometric method for determining surface orientation from multiple images. Opt Eng 19(1):139–144
2. Horn BKP (1975) Obtaining shape from shading information. The psychology of computer vision. McGraw-Hill, New York
3. Hayakawa H (1994) Photometric stereo under a light source with arbitrary motion. J Opt Soc Am A 11(11):3079–3089
4. Belhumeur PN, Kriegman DJ, Yuille AL (1999) The bas-relief ambiguity. Int J Comput Vis 35(1):33–44
5. Basri R, Jacobs D (2001) Photometric stereo with general, unknown lighting. In: IEEE conference on computer vision and pattern recognition (CVPR), Kauai, vol II, pp 374–381
6. Basri R, Jacobs D, Kemelmacher I (2007) Photometric stereo with general, unknown lighting. Int J Comput Vis 72(3):239–257
7. Basri R, Jacobs D (2001) Lambertian reflectance and linear subspaces. In: Proceedings of the IEEE international conference on computer vision, Vancouver, vol II, pp 383–390
8. Basri R, Jacobs D (2003) Lambertian reflectance and linear subspaces. IEEE Trans Pattern Anal Mach Intell 25(2): 218–233
9. Ramamoorthi R, Hanrahan P (2001) On the relationship between radiance and irradiance: determining the illumination from images of convex lambertian object. J Opt Soc Am A 18:2448–2459
10. Coleman EN, Jain R (1982) Obtaining 3-dimensional shape of textured and specular surfaces using four-source photometry. Comput Graph Image Process 18(4):309–328
11. Ikeuchi K (1981) Determining surface orientations of specular surfaces by using the photometric stereo method. IEEE Trans Pattern Anal Mach Intell 3(6):661–669
12. Georghiades A (2003) Incorporating the torrance and sparrow model of reflectance in uncalibrated photometric stereo. In: Proceedings of the IEEE international conference on computer vision, Nice, pp 816–823
13. Nayar SK, Ikeuchi K, Kanade T (1991) Shape from interreflections. Int J Comput Vis 6(13):173–195

14. Basri R, Frolova D (2008) A two-frame theory of motion, lighting and shape. In: IEEE conference on computer vision and pattern recognition (CVPR), Anchorage
15. Joshi N, Kriegman D (2007) Shape from varying illumination and viewpoint. In: Proceedings of the IEEE international conference on computer vision, Rio de Janeiro
16. Simakov D, Frolova D, Basri R (2003) Dense shape reconstruction of a moving object under arbitrary, unknown lighting. In: Proceedings of the IEEE international conference on computer vision, Nice, pp 1202–1207
17. Adato Y, Vasilyev Y, Zickler T, Ben-Shahar O (2010) Shape from specular flow. PAMI 32(11):2054–2070

# Photon, Poisson Noise

Samuel W. Hasinoff
Google, Inc., Mountain View, CA, USA

## Synonyms

Schott noise; Shot noise

## Related Concepts

▸Sensor Fusion

## Definition

Photon noise, also known as Poisson noise, is a basic form of uncertainty associated with the measurement of light, inherent to the quantized nature of light and the independence of photon detections. Its expected magnitude is signal dependent and constitutes the dominant source of image noise except in low-light conditions.

## Background

Image sensors measure scene irradiance by counting the number of discrete *photons* incident on the sensor over a given time interval. In digital sensors, the photoelectric effect is used to convert photons into electrons, whereas film-based sensors rely on photosensitive chemical reactions. In both cases, the independence of random individual photon arrivals leads to *photon noise*, a signal-dependent form of uncertainty that is a property of the underlying signal itself.

In computer vision, a widespread approximation is to model image noise as signal *independent*, often using a zero-mean additive Gaussian. Though this simple model suffices for some applications, it is physically unrealistic. In real imaging systems, photon noise and other sensor-based sources of noise contribute in varying proportions at different signal levels, leading to noise which is dependent on scene brightness. Understanding photon noise and modeling it explicitly is especially important for low-level computer vision tasks treating noisy images [2, 8] and for the analysis of imaging systems that consider different exposure levels [1, 4, 10] or sensor gains [5].

## Theory

Individual photon detections can be treated as independent events that follow a random temporal distribution. As a result, photon counting is a classic Poisson process, and the number of photons $N$ measured by a given sensor element over a time interval $t$ is described by the discrete probability distribution

$$\Pr(N = k) = \frac{e^{-\lambda t}(\lambda t)^k}{k!}\,, \tag{1}$$

where $\lambda$ is the expected number of photons per unit time interval, which is proportional to the incident scene irradiance. This is a standard Poisson distribution with a rate parameter $\lambda t$ that corresponds to the expected incident photon count. The uncertainty described by this distribution is known as *photon noise*.

Because the incident photon count follows a Poisson distribution, it has the property that its variance is equal to its expectation, $\mathrm{E}[N] = \mathrm{Var}[N] = \lambda t$. This shows that photon noise is signal dependent and that its standard deviation grows with the square root of the signal.

In practice, photon noise is often modeled using a Gaussian distribution whose variance depends on the expected photon count [1, 2, 4, 5, 8, 10],

$$N \sim \mathcal{N}(\lambda t, \lambda t)\,. \tag{2}$$

This approximation is typically very accurate. For small photon counts, photon noise is generally dominated by other signal-independent sources of noise, and for larger counts, the central limit theorem ensures that the Poisson distribution approaches a Gaussian.

Since photon noise is derived from the nature of the signal itself, it provides a lower bound on the uncertainty of measuring light. Even under ideal imaging conditions, free from all other sensor-based sources of noise (e.g., read noise), any measurement would still be subject to photon noise. When photon noise is the only significant source of uncertainty, as commonly occurs in bright photon-rich environments, imaging is said to be *photon limited*.

In general, the only way to reduce the effect of photon noise is to capture more signal. The ratio of signal to photon noise grows with the square root of the number of photons captured, $\sqrt{\lambda t}$. This shows that photon noise, while growing in absolute terms with signal, is relatively weaker at higher signal levels. However, in order to capture more photons, longer exposure times are required, and the number of photons captured in a single shot is limited by the full well capacity of the sensor. Note that while squeezed coherence lasers and other forms of nonclassical light can achieve amplitude noise below the photon noise limit [11], such exotic lighting configurations are typically not relevant for computer vision applications.

In digital sensors, a related source of noise that also follows a Poisson distribution is dark current noise. Dark current refers to "phantom" photon counts due thermal energy causing the sensor to release electrons at random. While photon noise is a property of the signal itself, dark current comes from the embodiment of the sensor and depends on both temperature and exposure time.

## Application

Photon noise is inherent to the measurement of light, has no parameters to be calibrated, and is independent of other noise sources. As a result, the effect of photon noise on imaging can be characterized using the *radiometric response function* that relates the photon count and the expected pixel intensity [3, 6].

To handle the signal dependence caused by photon noise, a first step is to estimate the noise variance for each pixel. This can be approximated in a straightforward way by inverting the forward model for imaging noise [5, 6, 9]. For increased accuracy, several other factors can be taken into account as well: the coupling between signal and noise leads to a recursive estimation [3]; pixels near saturation have reduced variance which can lead to bias [2, 8]; and on-camera processing such as demosaicking may introduce spatial correlation [8].

Image processing methods that explicitly incorporate more realistic signal-dependent models of noise, either calibrated [3, 5, 6] or inferred from the image [7, 8], adapt naturally to pixels of different intensities. As a result, for a variety of computer vision tasks such as denoising [3, 8] and edge detection [7], these methods can perform better than those handicapped by the assumption of scene-independent noise.

An alternative approach for handling signal-dependent noise is to transform the image using a *variable-stabilizing transformation* that amounts to applying per pixel nonlinearities that effectively reduce the signal dependence [2, 9]. Because the transformed signal approximates one with signal-independent noise, it may be processed using methods that assume a simpler noise model.

## References

1. Agrawal A, Raskar R (2009) Optimal single image capture for motion deblurring. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Miami, pp 2560–2567
2. Foi A, Trimeche M, Katkovnik V, Egiazarian K (2008) Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data. IEEE Trans Image Process 17(10):1737–1754
3. Granados M, Adjin B, Wand M, Theobalt C, Seidel H-P, Lensch Hendrik PA (2010) Optimal HDR reconstruction with linear digital cameras. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), San Francisco, pp 215–222
4. Hasinoff SW, Kutulakos KN, Durand F, Freeman WT (2009) Time-constrained photography. In: Proceedings of the IEEE international conference on computer vision, Kyoto, pp 333–340
5. Hasinoff SW, Durand F, Freeman WT (2010) Noise-optimal capture for high dynamic range photography. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), San Francisco, pp 553–560
6. Healey GE, Kondepudy R (1994) Radiometric CCD camera calibration and noise estimation. IEEE Trans Pattern Anal Mach Intell 16(3):267–276
7. Hwang Y, Kim J-S, Kweon I-S (2007) Sensor noise modeling using the Skellam distribution: application to the color edge detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Minneapolis, pp 1–8
8. Liu C, Szeliski R, Kang SB, Lawrence Zitnick C, Freeman WT (2008) Automatic estimation and removal of noise from a single image. IEEE Trans Pattern Anal Mach Intell 30(2):299–314
9. Prucnal PR, Saleh BEA (1981) Transformation of image-signal-dependent noise into image-signal-independent noise. Opt Lett 6(7):316–318

10. Treibitz T, Schechner YY (2009) Polarization: beneficial for visibility enhancement? In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Miami, pp 525–532
11. Vahlbruch H, Mehmet M, Chelkowski S, Hage B, Franzen A, Lastzka N, Goßler S, Danzmann K, Schnabel R (2008) Observation of squeezed light with 10-dB quantum-noise reduction. Phys Rev Lett 100(3):033602

## Picture Understanding

▶Line Drawing Labeling

## Piecewise Polynomial

▶Splines

## Pinhole Camera

▶Perspective Camera

## Pinhole Camera Model

Peter Sturm
INRIA Grenoble Rhône-Alpes, St Ismier Cedex,
France

### Related Concepts

▶Affine Camera; ▶Camera Calibration; ▶Center of Projection; ▶Focal Length; ▶Image Plane; ▶Optical Axis; ▶Weak Perspective Projection

### Definition

The pinhole camera model is the basic camera model used in computer vision. Its name originates from the concept of pinhole camera and it models perspective projections.

### Background

The pinhole model is the basic camera model used in computer vision. Its name stems from the concept of pinhole camera [1] (also related to the *camera obscura* [2]): usually, a closed box into which a single tiny hole is made with a pin, through which light may enter and hit a photosensitive surface inside the box (cf. Fig. 1). Pinhole cameras allow to take photographs of objects, which usually requires long exposure times due to the small aperture. The principles behind pinhole cameras and the *camera obscura* have been known, at least partially, since the fourth century BC [2].

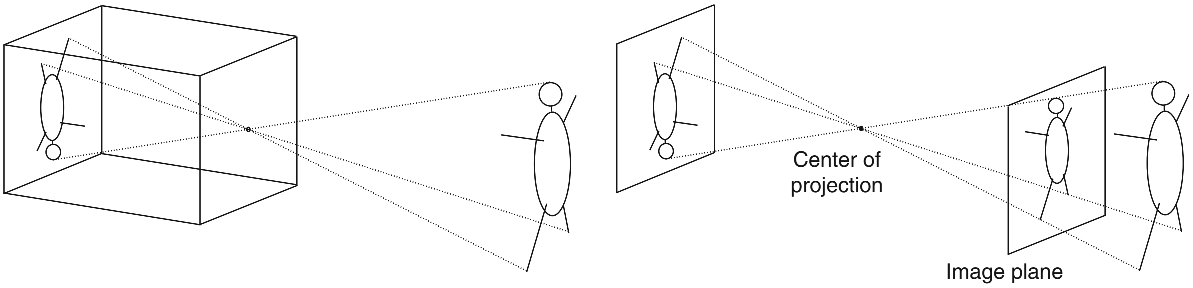
The pinhole camera model mimics the geometrical projection carried out by a pinhole camera, as follows (see also Fig. 1). The entire optics and aperture of a camera are reduced to a single point – the *optical center* or *center of projection*. The photosensitive surface is assumed to be planar and is, geometrically, represented by the so-called *image plane*. To determine where a 3D point is depicted in the image, it suffices to construct a straight line from that point and going through the optical center (one may consider this as a light ray). This line's intersection with the image plane gives the desired image point of the 3D point. A key property of this model is that points that are collinear in 3D get imaged to image points that are also collinear.

The pinhole model is simple and neglects many aspects of true cameras. For instance, apertures are finite and thus most 3D points are not imaged in a unique image point but within a finite area of the image plane. Likewise, pixels in digital cameras gather incoming light across finite areas. Thus, the pinhole model does not tell anything about blur, point spread functions, quantization, or other effects such as vignetting that occur in true cameras.

Other effects that are not modeled by it are radial or other geometric distortions, which usually occur with small focal lengths and that violate the above property of mapping collinear points in 3D to collinear points in the image. To handle such distortions, the pinhole model may be extended by adding models for radial distortion for instance, or by using different models altogether, such as becomes necessary for fisheye cameras. Despite the above limitations, the pinhole model is already a good approximation for many regular cameras.

### Theory

Above, the geometrical projection mapping described by the pinhole model is described. Before translating



**Pinhole Camera Model, Fig. 1** *Left*: sketch of a pinhole camera. *Right*: the two basic constituents of the pinhole camera model are the center of projection and the image plane. The true image plane shows inverse images of objects. To ease drawings, a common convention is to replace the true image plane by a virtual one between the center of projection and the scene, at the same distance from the center as the original image plane and parallel to it. This image plane produces an identical image, but that is "correctly" oriented

this into an algebraic formulation, a few notations are introduced. The *focal length* $f$ is the distance between the center of projection and the image plane. The line passing through the optical center and that is orthogonal to the image plane is called *optical axis*. The intersection point of the optical axis and the image plane is the *principal point*. These definitions are only based on the optical center and the image plane. When considering digital cameras, one in addition needs to take into account the layout of pixels in the image plane. The usual layout consists of a regular rectangular grid (although other arrangements, such as log-polar ones, were also experimented with [3, 4]): pixels are arranged into rows and columns. Let $k_u$ and $k_v$ be the column-wise and row-wise density of pixels, respectively (e.g., measured as number of pixels per millimeter). The value of $k_u/k_v$ is also called the *aspect ratio* of a camera. Usually, the two densities are equal to one another, that is, cameras have a unit aspect ratio, but especially with video cameras, this should not be taken for granted.

The following coordinate systems are used to derive algebraic expressions for the pinhole model (Fig. 2). The *camera coordinate system* has its origin in the optical center. A usual convention is to define the $Z$-axis as being coincident with the camera's optical axis and the $X/Y$-axes to be parallel to the columns and rows of pixels in the image plane. The *image coordinate system* has its origin in the principal point. The $x$-axis and $y$-axis are parallel to the $X$-axis and $Y$-axis, respectively, of the camera coordinate system. Typically, the camera and image coordinate systems use metric units. To represent the final digital image, one defines the *pixel coordinate system*. Its origin usually lies in one of the image area's corners. Let its coordinates, relative to the image coordinate system, be $(-x_0, -y_0)$. The two axes, $u$ and $v$, are parallel to $x$ and $y$, respectively; their unit is (number of) pixels, counted row-wise and column-wise, respectively. Note that the above choices of coordinate systems are not unique; other choices are possible, for example, for the $X$ and $Y$ axes and the origin of the pixel coordinate system, although care should be taken to use right-handed systems. The following equations will have to be adapted accordingly.

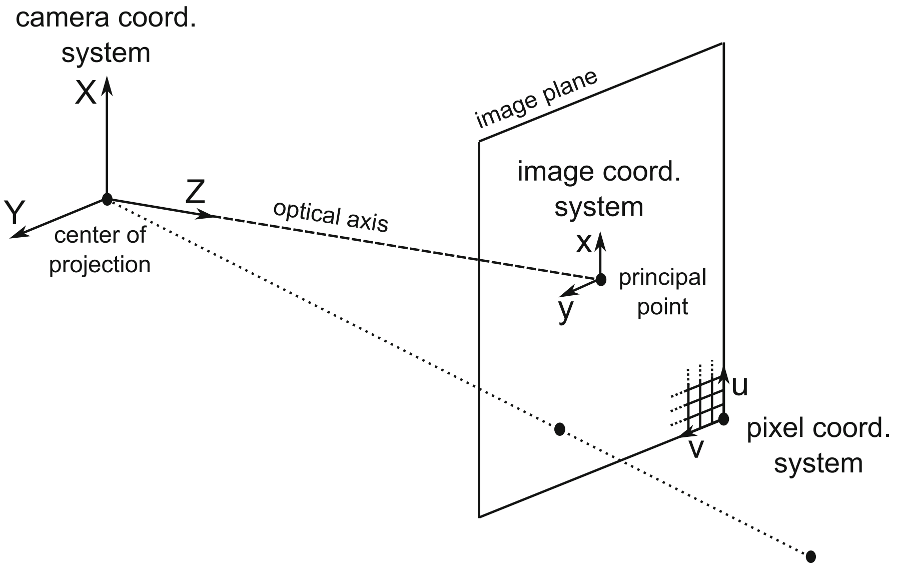
With the above definitions, the projection carried out by the pinhole camera model can be formulated as follows. Let $(X, Y, Z)$ be the coordinates of a 3D point, expressed in the camera coordinate system. From the comparison of similar triangles, one obtains the image point's coordinates in the image coordinate system as:

$$x = f\frac{X}{Z} \quad y = f\frac{Y}{Z}.$$

The coordinates in the pixel coordinate system are obtained by applying a translation corresponding to the shift of origin and scalings corresponding to the conversion from metric coordinates to pixel ones:

$$\begin{aligned} u &= k_u(x + x_0) = k_u f\frac{X}{Z} + k_u x_0 \\ v &= k_v(y + y_0) = k_v f\frac{Y}{Z} + k_v y_0. \end{aligned}$$

It is often useful to express these projection equations using homogeneous coordinates for the 3D and image points:



**Pinhole Camera Model, Fig. 2** The components of the pinhole camera model. The distance between the center of projection and the image plane is the focal length $f$

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \begin{pmatrix} k_u f & 0 & k_u x_0 & 0 \\ 0 & k_v f & k_v y_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \qquad (1)$$

where $\sim$ signifies equality up to scale of vectors or matrices.

It is common to replace the above “metric” entities $f, k_u, k_v, x_0$, and $y_0$ by equivalent ones given in pixel units:

$$\alpha_u = k_u f \quad \alpha_v = k_v f \quad u_0 = k_u x_0 \quad v_0 = k_v y_0.$$

Here, $\alpha_u$ and $\alpha_v$ measure the focal length in number of pixels (column-wise and row-wise, respectively) and $(u_0, v_0)$ are the coordinates of the principal point given in the pixel coordinate system. These four entities are also called the *intrinsic parameters* of the pinhole camera model, since they describe what happens “inside” a camera. They are often grouped together in an upper triangular so-called *calibration matrix*:

$$\mathsf{K} = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}. \qquad (2)$$

Sometimes, a fifth intrinsic parameter is added to the model – a so-called skew parameter that replaces the zero in the first row of the calibration matrix and which allows to model pixel layouts with skewed axes or cameras with desynchronized pixel readout. With modern digital cameras, these issues can usually be neglected though.

In order to model cameras in motion or multi-camera systems, one needs to describe the position and orientation of a camera. To do so, a final coordinate system, the *world coordinate system*, is considered. This may be attached to a physical object, for instance, an object to be inspected; otherwise, it can be assumed arbitrarily, as long as it remains fixed throughout an application. Let $\mathbf{t}$ be the coordinates of the center of projection in the world coordinate system and let the rotation matrix $\mathsf{R}$ represent the camera's orientation. Then, a 3D point is mapped from the world to the camera coordinate system, as:

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} \mathsf{R} & -\mathsf{R}\mathbf{t} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{pmatrix} \begin{pmatrix} X^w \\ Y^w \\ Z^w \\ 1 \end{pmatrix}, \qquad (3)$$

where as above, homogeneous coordinates are used.

Putting together (Eq. 1) and (Eq. 3), we get the complete expression of the pinhole camera model:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \begin{pmatrix} k_u f & 0 & k_u x_0 & 0 \\ 0 & k_v f & k_v y_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \mathsf{R} & -\mathsf{R}\mathbf{t} \\ \mathbf{0}^{\mathsf{T}} & 1 \end{pmatrix} \begin{pmatrix} X^w \\ Y^w \\ Z^w \\ 1 \end{pmatrix}.$$

With the above definitions of intrinsic parameters and calibration matrix (Eq. 2), this can be written more compactly as:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \underbrace{\mathsf{KR}\begin{pmatrix} \mathsf{Id}_3 & -\mathbf{t} \end{pmatrix}}_{\mathsf{P}} \begin{pmatrix} X^w \\ Y^w \\ Z^w \\ 1 \end{pmatrix},$$

where $\mathsf{Id}_3$ is the $3 \times 3$ identity matrix. The $3 \times 4$ matrix $\mathsf{P}$ is called *projection matrix* or *camera matrix*.

The projection expressed by the pinhole model is a perspective projection. Even simpler camera models exist in the form of orthographic or other affine projections.

## Application

The pinhole camera model, like any other camera model [5], can be used to infer geometrical information about an imaged scene, from one or more images. Some of the most common applications are sketched below. It is usually assumed that a camera is calibrated prior to an application, that is, that its intrinsic parameters are known through a *camera calibration* procedure. Most camera calibration procedures utilize a reference object of known shape, a calibration grid. However, there also exist approaches for *self-calibration* (or autocalibration or on-line calibration) that allow to compute the intrinsic parameters directly from images of an unknown scene.

*Pose estimation* refers to the computation of an object's position and orientation relative to a camera, or vice-versa. Here, it is usually assumed that the object's shape is known and the camera calibrated. *Motion estimation* is the determination of the camera's motion between two or more acquisitions; this is possible even if the scene is unknown, that is, if it does not contain any reference object of known shape or type. Motion estimation usually requires some amount of image matching, that is, the determination of projections of the same scene feature in different images. *3D modeling* is usually done from two or more images of a rigid scene and also usually requires image matching, although exceptions exist, such as shape-from-shading, photometric stereo, and interactive single-view 3D modeling.

Note that none of these applications is specific to the pinhole model, although the theory underlying them has most extensively been studied for this model, as parts of perspective multi-view geometry [6]. It is also reminded, as said above, that the pinhole model neglects many aspects of image formation and does not model non-perspective image distortions. A camera model, be it the pinhole or another one, should only be used in an application if it is sure that it is appropriate for the camera used.

## References

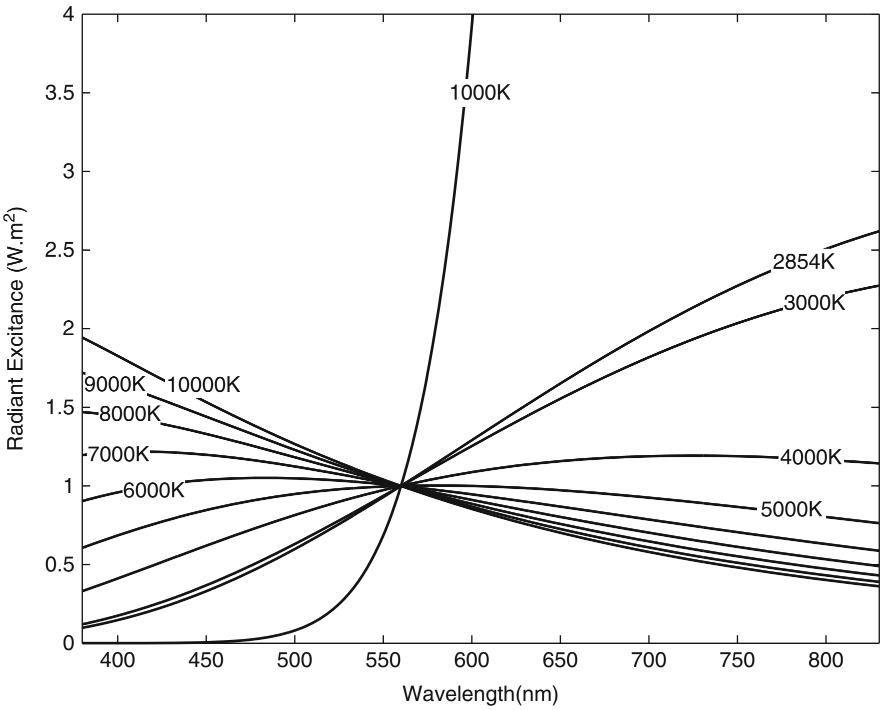
1. Wikipedia (2011) Pinhole camera. http://en.wikipedia.org/wiki/Pinhole_camera. Accessed 3 Aug 2011
2. Wikipedia (2011) Camera obscura. http://en.wikipedia.org/wiki/Camera_obscura. Accessed 5 Aug 2011
3. Tistarelli M, Sandini G (1993) On the advantage of polar and log-polar mapping for direct estimation of time-to-impact from optical flow. IEEE Trans Pattern Anal Mach Intell 15(4):401–410
4. Pardo F, Dierickx B, Scheffer D (1997) CMOS foveated image sensor: signal scaling and small geometry effects. IEEE Trans Electron Devices 44(10):1731–1737
5. Sturm P, Ramalingam S, Tardif JP, Gasparini S, Barreto J (2011) Camera models and fundamental concepts used in geometric computer vision. Found Trends Comput Graph Vis 6(1–2):1–183
6. Hartley R, Zisserman A (2004) Multiple view geometry in computer vision, 2nd edn. Cambridge University Press, Cambridge

# Planckian Locus

Rajeev Ramanath[1] and Mark S. Drew[2]
[1]DLP® Products, Texas Instruments Incorporated,
Plano, TX, USA
[2]School of Computing Science, Simon Fraser
University, Vancouver, BC, Canada

## Synonyms

Blackbody radiator; Thermal radiator



**Planckian Locus, Fig. 1** Spectral power distributions of various blackbody radiators from 1,000 to 10,000 K, with all spectral distributions normalized to unity at 560 nm

## Definition

The Planckian locus as it relates to color is the locus of points in a color space that would be followed by an incandescent blackbody radiator as its temperature changes. This locus is typically described in the CIE $x, y$ or CIE $u', v'$ chromaticity spaces.

## Background

The CIEXYZ color space is defined by

$$\begin{aligned} X &= k \int_\lambda \bar{x}(\lambda) i(\lambda) r(\lambda) d\lambda \\ Y &= k \int_\lambda \bar{y}(\lambda) i(\lambda) r(\lambda) d\lambda \\ Z &= k \int_\lambda \bar{z}(\lambda) i(\lambda) r(\lambda) d\lambda, \end{aligned} \tag{1}$$

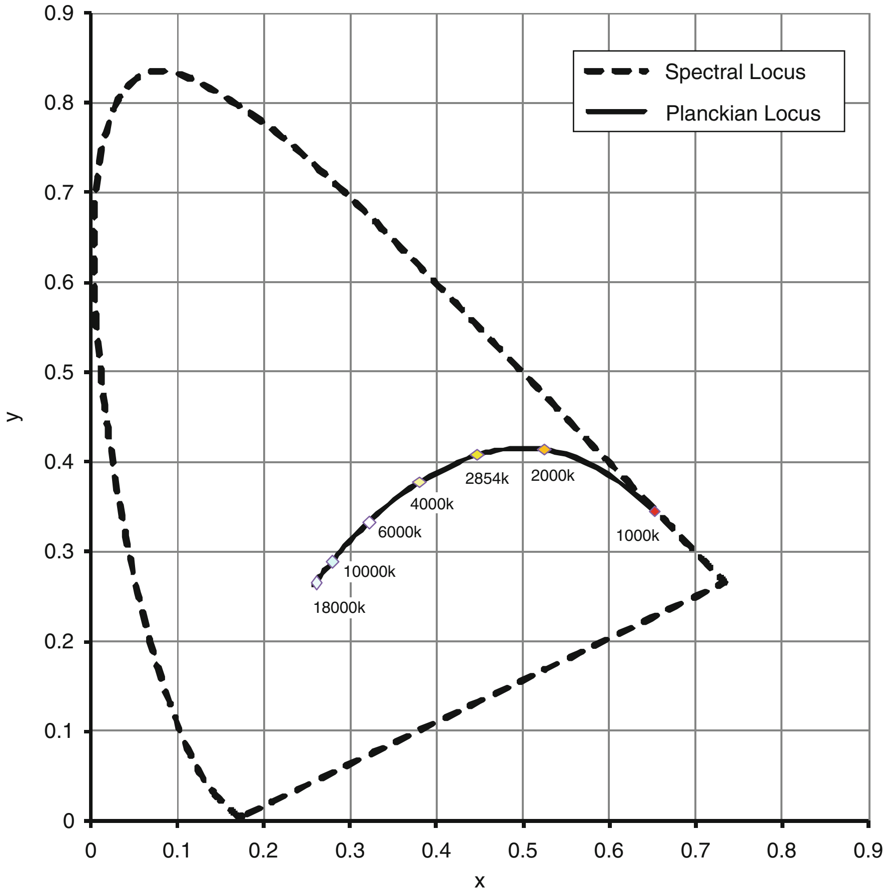
where $k$ denotes a normalization factor that is set to 683 lumens/Watt in the case of absolute colorimetry and to $100/\int_\lambda \bar{y}(\lambda) i(\lambda) d\lambda$ for relative colorimetry and $i(\lambda)$ denotes the spectrum of an illuminant. The functions $\bar{x}(\lambda), \bar{y}(\lambda), \bar{z}(\lambda)$ are the CIE color-matching functions.

In the case of relative colorimetry, this means that a value of $Y = 100$ denotes the brightest color – the illuminant reflecting from a perfect reflecting diffuser [1, 3].

Planck's radiation law describes the spectral distribution of radiant excitance $M_e$ as a function of wavelength $\lambda$ and temperature $T$ and is given by Planck's Law:

$$M_e(\lambda, T) = \frac{c_1}{\lambda^5 \left[ \exp\left(\frac{c_2}{\lambda T}\right) - 1 \right]} \tag{2}$$

where $c_1 = 2\pi h c^2 = 3.74183 \times 10^{-16} W.\text{m}^{-2}$, $c_2 = h.c/k = 1.4388 \times 10^{-2}\text{m.K}$ ($c$ is the speed of light in vacuum, $h$ is Planck's constant, $k$ is Boltzmann's constant), and the excitance is defined in units of $W.\text{m}^{-3}$.



**Planckian Locus, Fig. 2** Projection of the emission spectra of various blackbody radiators on the CIE $x, y$ chromaticity diagram showing the locus of spectral colors along with the Planckian locus

Radiation emitted from blackbody radiators is defined by Planck's Law and is among the few radiations from sources that has its relative spectral power distribution match those of illuminants. In other words, blackbody radiators are the select few sources of illumination that match standard illuminant spectral power distributions – this may be seen as valid in the case of the equivalence of standard illuminant "A" and a blackbody with a temperature 2,856 K. Radiation from these radiators is typically seen as "white" to human observers. This makes it important to plot a locus of these points in a space such as CIEXYZ, or shown in a simpler view such as the two-dimensional space of the CIE $x, y$ chromaticity diagram. The locus of points followed by illuminants defined by Planck's Law is called the Planckian locus. Figure 1 shows the spectral power distribution of various blackbody radiators from 1,000 to 10,000 K, with all spectral power distributions normalized to unity at 560 nm. As the blackbody gets hotter ($T$ increases), one can see that the red content in the spectrum reduces and the blue content increases – an indication of the color as would be seen by the human observer.

The projection of the emission spectra of various blackbody radiators onto the CIEXYZ color space via Eq. (2), and then further projected onto the CIE $x, y$ 2-D chromaticity diagram, is shown in Fig. 2, for the CIE 2-degree observer [1]. One can see that the cooler blackbody radiators (lower $T$) are more red in their appearance (as given by their location in the chromaticity diagram) and the hotter blackbody radiators (higher $T$) are more blue in their appearance – the chromaticity diagram is red at the bottom right, green at the top, and blue at the bottom left.

## References

1. CIE 15:2004 (2004) Colorimetry. CIE, Vienna
2. 1998c CIE (1998) CIE standard illuminants for colorimetry. CIE, Vienna. Also published as ISO 10526/CIE/S006/E1999
3. Wyszecki G, Stiles WS (1982) Color science: concepts and methods, quantitative data and formulas, 2nd edn. Wiley, New York

# Plane Sweeping

David Gallup[1] and Marc Pollefeys[2]
[1]Google Inc., Seattle, WA, USA
[2]Computer Vision and Geometry Lab (CVG) –
Institute of Visual Computing, Department of
Computer Science, ETH Zürich, Zürich, Switzerland

## Related Concepts

▶Multi-baseline Stereo

## Definition

Plane sweeping is a multi-view stereo algorithm notable for its efficiency, especially on modern graphics hardware (GPU).

## Background

Plane sweeping addresses the stereo problem, which is to find the surface of a scene, given two or more calibrated views of the scene. Assuming that surfaces are Lambertian and that there are no occlusions, a point on the surface will have the same appearance in all views. Therefore, a conceptual solution to the stereo problem is to find the points that maximize *photoconsistency*. This can be done by searching along the viewing rays for some image. The distance to the surface along the ray is called *depth*, and the set of all depths is called a depth map. Plane sweeping is a technique for efficiently organizing this search by sweeping a plane through space. Projecting all points on a plane into a perspective camera is simply a homography transformation, which can be executed efficiently on modern graphics hardware (GPU) [8].

Binocular stereo would typically *rectify* a pair of images such that searching along a viewing ray is equivalent to looping over the pixels in a row of the other image [6]. But three or more views, in general, cannot be rectified simultaneously. Plane sweeping avoids this problem by operating in 3D, searching along rays, one plane at a time, instead of searching image pixels. However, the proper sampling of 3D space is an issue. The image signal is sampled in 2D image space, and the sampling of the plane sweep must map to a similar sampling rate in image space. The right sampling is important for both correctness and efficiency.

## Theory

The input to the plane sweep algorithm is a set of *views*. Let a view be defined as an image plus camera parameters. For simplicity, choose one view to be a *reference view*. All other images will be compared against the reference image to measure photoconsistency. The output of the algorithm will be a depth map for the reference view. Let the reference camera be identity, that is, its camera matrix is

$$\mathbf{P}_{\text{ref}} = \mathbf{K}_{\text{ref}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{1}$$

Let $\mathcal{I} = \{I_1, I_2, \ldots, I_N\}$ be a set of matching views whose cameras are $\mathcal{P} = \{\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_N\}$, where

$$\mathbf{P}_i = \mathbf{K}_i \begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \end{bmatrix}. \tag{2}$$

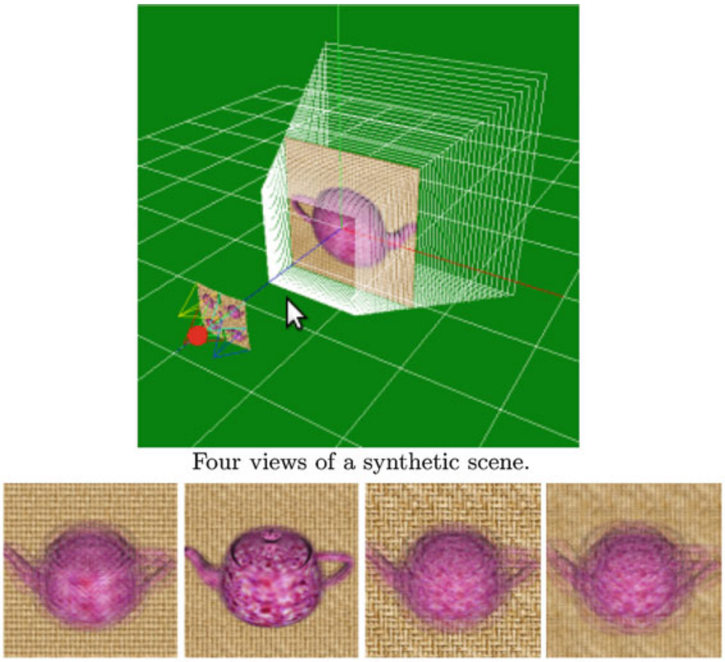
Let a plane $\pi$ be defined as

$$\pi = \begin{bmatrix} \mathbf{n}_x & \mathbf{n}_y & \mathbf{n}_z & d \end{bmatrix}^\top. \tag{3}$$

The plane can be swept through space in direction $\mathbf{n}$ by varying $d$. For a given plane, photoconsistency will be evaluated on the plane at each point. This can be accomplished by projecting all images onto the plane and comparing the image values. Ultimately, it is desired to compute photoconsistency in the image to be reconstructed, and so, each image will be projected onto the plane and then into reference image. This transformation is known as a plane homography. The equation for the plane homography from $I_{\text{ref}}$ to $I_i$ via the plane $\pi$ is

$$\mathbf{H}_i^\pi = \mathbf{K}_i \left( \mathbf{R}_i - \frac{\mathbf{t}_i \mathbf{n}^\top}{d} \right) \mathbf{K}_{\text{ref}}^{-1}. \tag{4}$$

**Plane Sweeping, Fig. 1** Plane sweep of a synthetic scene. As the plane sweeps from front to back, you can see the alignment of the teapot followed by the mat behind it



Four views of a synthetic scene.

The warped images blended together at different stages of the plane sweep.

The homography $\mathbf{H}_i^{\pi}$ maps from the reference image to the image $i$ so that pixel $(x, y)$ in the warped image can be computed as

$$\begin{aligned}\tilde{I}_i(x,y) &= I\left(\frac{\tilde{x}}{\tilde{w}}, \frac{\tilde{y}}{\tilde{w}}\right) \text{ where } \begin{bmatrix}\tilde{x} & \tilde{y} & \tilde{w}\end{bmatrix}^{\top} \\ &= \mathbf{H}_i^{\pi}\begin{bmatrix}x & y & w\end{bmatrix}^{\top}. \qquad (5)\end{aligned}$$

The photoconsistency at any point in the reference view can be computed using any number of matching scores. Common choices include the sum of squared differences (SSD), sum of absolute differences (SAD), and normalized cross correlation (NCC). These scores are window-based because they operate over a neighborhood or correlation window. This produces a more robust matching score at the expense of some overextension artifacts [3]. As an example, the SSD can be computed as follows:

$$\begin{aligned}SSD(x,y) = \sum_{i=1}^{N} \sum_{(i,j)\in\mathcal{N}} &(I_{\text{ref}}(x+i, y+j) \\ &-\tilde{I}_i(x+i, y+j))^2. \qquad (6)\end{aligned}$$

The SSD is actually a photoconsistency *cost*, and it will be *minimized* by points near the surface. Defining the neighborhood $\mathcal{N}$ in the space of the reference image assumes that points in the neighborhood are on the plane $\pi$. For scenes with highly slanted surfaces, better results can be achieved by sweeping the plane through space in different directions (different **n**). Searching all directions is slow, but in some cases, like man-made scenes, a few dominant directions are sufficient [1].

Equation 5 assumes that all matching views are unoccluded, but this is often not the case, and an occluded view can produce an arbitrarily high matching cost. Because plane sweeping can use many views, occlusion handling amounts to summing over some subset of views likely to be unoccluded. In the case where the camera path is close to linear, the subset can be either the previous or the subsequent half-set of matching views. A more general solution is to assume the best 50% of scores are unoccluded [4].

While sweeping the plane through space, the plane with the best photoconsistency score is recorded per pixel. The depth for each pixel can be computed by intersecting the viewing ray with the recorded best plane. The warped images at various positions in the plane sweep are shown in Fig. 1.

Proper sampling of the 3D space during the plane sweep is important for both correctness and efficiency. Sampling too sparsely could miss the photoconsistency optimum, and sampling too densely is inefficient. During the plane sweep, the plane should move so that the motion of the warped images is no more than 1 pixel (or the desired image sampling rate). Not all pixels move at the same rate, but measuring the corner pixels of the reference image is sufficient. Since the planar warps are linear, the interior points are linear combinations of corners and are therefore bounded. Sampling can be sped up by using downsampled images, effectively making pixels larger. Varying the resolution as well as varying the set of matching images (to control the baseline) can be used to control the sampling rate to achieve an optimal balance between efficiency and precision [2].

## Application

The plane sweep algorithm has been popular for several real-time and large-scale systems, where efficiency is a key concern. The Urbanscape system is able to process VGA resolution video at over 30 frames per second, allowing it to reconstruct entire cities from street-level video in a matter of hours [1, 5].

A plane sweep approach was used in [7] for real-time view synthesis which allowed virtual views of the scene to be rendered from camera positions that were not physically captured. This algorithm differs slightly from plane sweep stereo in that the most consistent color, rather than the depth, is returned.

## References

1. Gallup D, Frahm J-M, Mordohai P, Qingxiong Y, Pollefeys M (2007) Real-time plane-sweeping stereo with multiple sweeping directions. In: Computer vision and pattern recognition (CVPR). IEEE, Piscataway, Minneapolis, Minnesota
2. Gallup D, Frahm J-M, Mordohai P, Pollefeys M (2008) Variable baseline/resolution stereo. In: Computer vision and pattern recognition (CVPR). IEEE, Piscataway, Anchorage, Alaska
3. Kanade T, Okutomi M (1994) A stereo matching algorithm with an adaptive window: theory and experiment. In: Pattern analysis and machine intelligence (PAMI), IEEE transaction; IEEE Trans Pattern analysis and machine intelligence, 16(9):920-932
4. Kang SB, Szeliski R, Chai J (2001) Handling occlusions in dense multi-view stereo. In: Computer vision and pattern recognition (CVPR). IEEE Computer Society, Los Alamitos, Kauai, Hawaii, pp I:103–110
5. Pollefeys M, Nistâer D, Frahm J-M, Akbarzadeh A, Mordohai P, Clipp B, Engels C, Gallup D, Kim S-J, Merrell P, Salmi C, Sinha S, Talton B, Wang L, Yang Q, Stewâenius H, Yang R, Welch G, Towles H (2008) Detailed real-time urban 3d reconstruction from video. Int J Comput Vis 78: 143–167
6. Scharstein D, Szeliski R (2002) A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int J Comput Vis 47:7–42
7. Yang R, Pollefeys M (2002) Real-time consensus-based scene reconstruction using commodity graphics hardware. In: Pacific graphics. Academic, San Diego
8. Yang R, Pollefeys M (2003) Multi-resolution real-time stereo on commodity graphics hardware. In: International conferences on computer vision and pattern recognition (CVPR), Madison, Wisconsin, pp I:211–217

# Plenoptic Function

Shing Chow Chan
Department of Electrical and Electronic Engineering,
The University of Hong Kong, Hong Kong, China

## Synonyms

Image-based rendering (IBR)

## Related Concepts

▶Light Field; ▶Lumigraph

## Definition

The plenoptic function describes the intensity of each light ray in the world as a function of visual angle, wavelength, time, and viewing position.

## Background

The term plenoptic was derived from the word roots plen- (plenus) and opti- (optos), which means full/complete and eye/view, respectively. The plenoptic function was coined by Bergen and Anderson [1] to describe the intensity of each light ray in the world as a function of visual angle, wavelength, time, and viewing position. It captures everything that can potentially be seen by an optical device and is related to previous concept of J. J. Gibson's "the structure of ambient light" and Leonardo da Vinci's "visual pyramid."

A light ray in the space can be parameterized by a position with three dimensions and a direction or visual angle in two dimensions (Fig. 1). Therefore, together with the wavelength and time dimensions, the plenoptic function is a seven dimensional (7D) function. The study of early vision is thus closely related to the sampling and processing of the plenoptic function. For instance, the derivatives of the plenoptic function with the position and time give usual information regarding the motion of objects, etc. In computer vision and video processing, the wavelength domain is simplified by sampling in the $(R, G, B)$ color system. Consequently, images and videos are just two-dimensional (2D) and three-dimensional (3D) special cases or samples of the 7D plenoptic function. Based on this function, theoretically, novel views at different positions and time can be reconstructed from its samples, provided that the sample rate is sufficiently high. Because of the multidimensional nature of the plenoptic function, various such simplifications called image-based representations have been proposed to render new views from the representations with different complexity/functionalities trade-offs. This is the foundation of image-based rendering (IBR) without using geometry, and it usually requires large amount of samples.

As the plenoptic function observed is a consequence of the interaction of the light sources with objects in the scene, which further involve their geometries and surface properties, there is considerable redundancy in the plenoptic function, which can be further exploited by estimating or measuring the geometry, lightings, and surface properties of the scene to different extent. Image-based representations employing such modeling approach and auxiliary information, called image-based modeling, generally will provide improved user interaction and require fewer samples for rendering. The capturing, sampling, rendering, and processing of the plenoptic function are important areas of research in IBR and related applications such as computational photography, 3D/multiview videos and displays, etc.

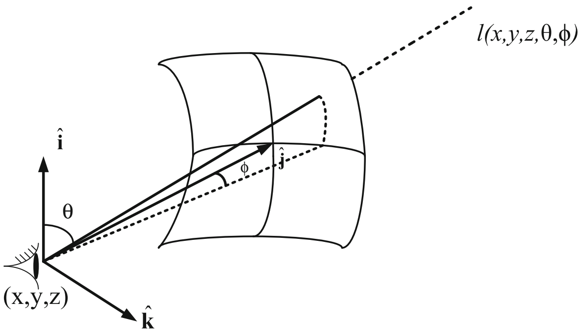
## Theory

The amount or intensity of light along a ray is measured in radiance which is the power transmitted per unit area perpendicular to the direction of travel, per unit solid angle. Therefore, the plenoptic function that describes the radiance along light rays traveling in every direction through every point, denoted by $l$, is in watts (W) per meter squared (m$^2$) per steradian (sr) (W/(m$^2$sr)).

The 7D plenoptic function can be parameterized using different coordinate systems, such as the familiar Cartesian, spherical, or cylindrical coordinates system. For instance, if the position $V$ is parameterized by the Cartesian coordinates $(x, y, z)$ while the direction is parameterized by the spherical coordinate $(\theta, \phi)$ where $\theta$ and $\phi$ are the elevation and azimuth angles, respectively, as shown in (Fig. 1), the plenoptic function can be written as $l(x, y, z, \theta, \phi, \lambda, \tau)$ where $\lambda$ and $\tau$ denote respectively the wavelength and time. By employing different parameterization and simplification, different image-based representations can be derived from the plenoptic function.

### Representation

The conventional camera employs a lens to compress the light rays passing through an opening called the aperture and obtain a 2D image by placing an array of imaging sensors at the focal plane of the lens. For static scene, one can rotate the camera along the camera center and capture the rays at a given position $V$ with different elevation and azimuth angles. The plenoptic function is then reduced to a panorama $l_V(\theta, \phi)$ with two dimensions. Since the sensor array is usually rectangular in shape, some rebinning or stitching of the images is necessary [5, 19]. Moreover, due to the finite field of view of cameras, the samples may be incomplete near the two poles of the sphere. This spherical set of rays can also be projected on a cube or a cylinder, which provides other convenient representation of panoramas. A panoramic video can be obtained by employing multiple closely spaced video cameras, instead of rotating a single camera, to obtain a 3D plenopic function $l_V(\theta, \phi, \tau)$ for dynamic scenes. The close relationship between plenoptic function and image-based rendering was due to McMillan and Bishop [14] who proposed plenoptic modeling



**Plenoptic Function, Fig. 1** Plenoptic function as a light energy received

using the 5D complete plenoptic function for static scene $l(x, y, z, \theta, \phi)$.

In the free space, the radiance along rays remains constant. This one-dimensional redundancy in the plenoptic function allows us to reduce it to a 4D function for static scene, which is called the light field [11] and lumigraph [8] in computer graphics. A similar concept in video communications is called the ray space [7]. The collection of light rays in a 4D static light field can be parameterized in a number of ways. A commonly used parameterization is the two-plane parameterization, where a light ray in the light field is parameterized as its intersections or coordinates with two parallel planes. These rays can be captured by taking a series of pictures on a 2D rectangular plane, which results in an array of images. Lumigraph differs from light field in that geometry in form of depth map is utilized to improve the rendering quality, which paves the way to more sophisticated representations in image-based modeling. For more information, see the section on light field and lumigraph. In [17], an outward facing camera moving on a circle was used to capture a series of densely sampled images, called concentric mosaic, of a static scene. This gives rise to a 3D representation, which can be used to render views inside the circle. A brief summary of these classical representations is given in Fig. 2. See the section on IBR for more illustration. These parameterizations can be further simplified by restricting the camera locations to line, line segments [4, 23], circle, circular arc, etc. to reduce hardware complexity of the capturing system. This gives rise to a wide range of image-based representations specified by different camera geometry [18]. For time-varying or dynamic scenes, similar parameterization can be employed. However, light rays at the viewing locations have to be captured continuously. This can be done by an array of video cameras or specially design capturing devices.

The plenoptic function of static scene has also been extended to include the change of illumination direction as in the plenoptic illumination function [20] at the expenses of increased data samples to be recorded at different lighting directions. Since the effect of multiple light sources is additive, one can relight a panorama or light field with arbitrary lightings using such representations.

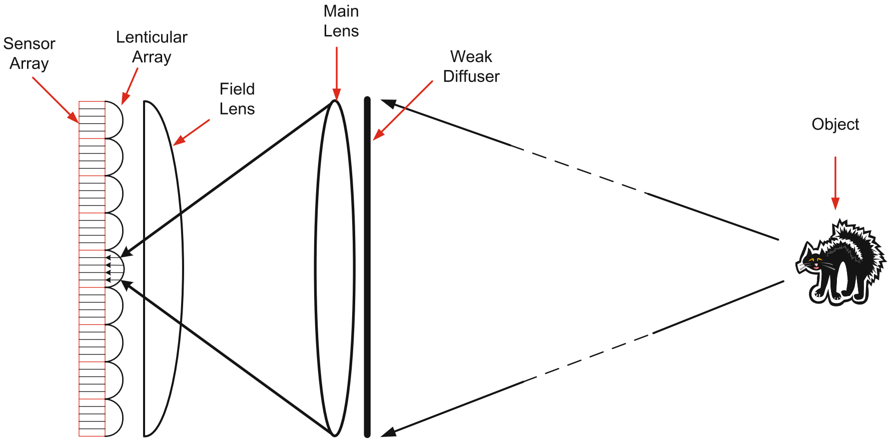
## Plenoptic Cameras

The term plenoptic camera was coined by Anderson and Wang [2]. It is a camera that captures a chunk of the optical structure of the light impinging on the lens. Basically, it records information about how the world appears from all possible viewpoints within the lens aperture. The plenoptic camera in [2] collects light with a single lens, but it uses a microlens or lenticular array at the image plane to redirect lights at certain directions at this location as collected from the lens aperture onto the sensor array (Fig. 3). Effectively, the plenoptic camera captures the cylindrical or spherical light fields over the lens aperture as macropixels onto a sensor array, depending respectively on whether a cylindrical or spherical lenticular array is used. The same principle has been used to produce autostereoscopic/multiview 3D display [10]. As a ray has to be mapped to an imaging pixel and there may be several directions to be recorded in a macropixel, the effective resolution of sensor array will be reduced. In [2], a (5 × 5) macropixel is employed. One can therefore expect such resolution reduction will be more severe for spherical than cylindrical lenticules.

Using the rays or light field recorded by the plenoptic camera, one can render novel views by arranging the light rays captured as in light field rendering. The images captured by a cylindrical lenticules is a 3D light field which support viewpoint changes along a horizontal line, whereas a spherical lenticules is able to capture a 4D light field as demonstrated and refined later by Ng et al. [16]. Moreover, as mentioned in [2], one can measure the parallax corresponding to these virtual displacements and derive depth estimates for objects in the scene. Since the virtual displacement, and hence the potential resolution of depth, is somewhat limited by the size of the lens aperture, it tends to be lower than using stereo cameras.

**Plenoptic Function, Fig. 2** A taxonomy of plenoptic functions

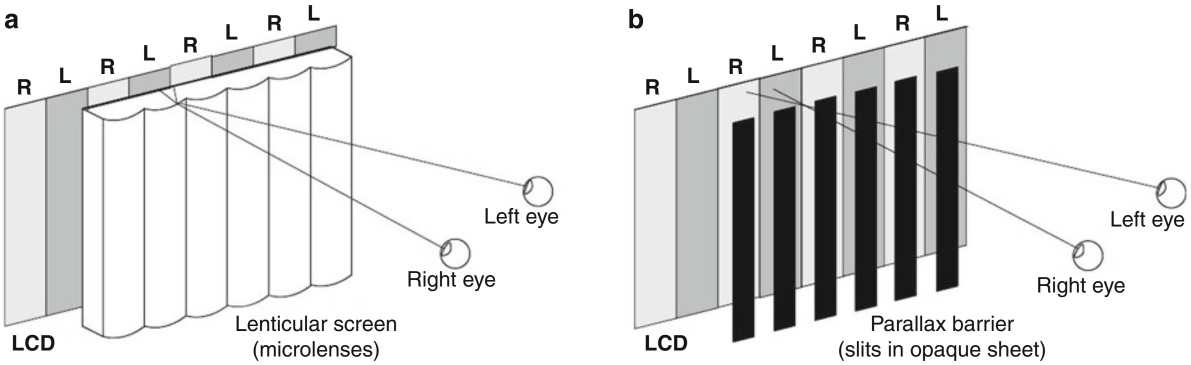
| Dimension | Year | View space | Name |
|---|---|---|---|
| 7 | 1991 | Free | Plenoptic function |
| 5 | 1995 | Free | Plenoptic modeling |
| 4 | 1996 | Bounding box | Lightfield/Lumigraph |
| 3 | 1999 | Bounding circle | Concentric Mosaics |
| 2 | 1994 | Fixed point | Cylindrical/Spherical panorama |



**Plenoptic Function, Fig. 3** Plenoptic camera employing lenticular lens [2]

The concept of inserting optical elements or masks in the camera for 3D imaging dates back nearly a century ago, called integral photography or parallax panoramagrams and realized using a fly-eye lens array or a slit plate [9, 13]. Basically, these manipulate the 4D light field spectrum by modulation or reparameterization to make it fit into a 2D sensor array [2]. Another technique is to make use of coded aperture or shutter as an optical modulator to capture stereoscopic images [6] and more recently to preserve the high-frequency components of motion-blurred images and to provide high-dynamicrange. Another recent technique is to employ multiple capturing of the scene sequentially by using beam splitters and camera arrays where at each exposure, the image parameters such as lighting, exposure time, focus, viewpoints, or spectral sensitivity are made different. After processing, a quality image or additional information is obtained. In [12], a coded aperture implemented using a programmable liquid crystal array, and multiple exposure was employed to capture light field using a single camera and derive the corresponding depth information.

Early systems for capturing light fields or plenoptic function for large environmental model usually involves camera arrays [4, 15, 21–23]. They are more expensive and difficult to build, as compared with a single plenoptic camera. On the other hand, due to their large baseline, it can also support a larger range of possible view points.



**Plenoptic Function, Fig. 4** Multiview display employing (**a**) lenticular array and (**b**) parallax barrier

## Sampling and Compression

During the capturing and display of plenoptic function, it is important to determine the number of samples required and how to obtain a satisfactory rendering quality. This problem was first studied for the light fields in [3] using the concept of Fourier transform. For Lambertian surfaces and rectangular sampling (see the section on light field for a more detailed treatment), it was found that the maximum camera spacing in the $v$ plane is $\Delta t_{\max} = 1/(2\pi K_{\Omega_v} f(\frac{1}{z_{\min}} - \frac{1}{z_{\max}}))$ (assuming the notation of the lumigraph), where $z_{\min}$ and $z_{\max}$ denote respectively the minimum and maximum depth values of the scene and $K_{\Omega_v}$ is the maximum frequency of the light field in the $v$ plane, which depends on the maximum frequency of texture variations $B_v$, the resolutions of the sampling camera $1/\Delta v$, and the rendering camera $1/\delta v$ as $K_{\Omega_v} = \min(B_v, 1/(2\Delta v), 1/(2\delta v))$. Similar results hold for the s plane. To reduce the sampling rate and avoid dense sampling, the value of $K_{\Omega_v}$ can be decreased by reducing $B_v$ and $1/(2\Delta v)$, through prefiltering the light field images, to the desired rendering resolution. A more thorough discussion on the effect of geometry on this minimum sampling density or rate can be found in [18, 22]. It was found that the sampling rate can be reduced by decomposing the light fields into depth layers. Therefore, recent research has focused on the estimation of geometry of objects using stereo or multiview vision techniques, special depth sensing devices, and lighting techniques as in photometric stereo.

Since the plenoptic function is a high-dimensional and highly correlated signal, they need to be compressed for efficient storage and transmission. Comprehensive reviews of the subjects are available at [18]. See also the sections on image-based rendering and light field.

## Multiview Displays

Multiview three-dimensional displays consist of view-dependent pixels that reveal a different color according to the viewing angle and offer viewing of simple plenoptic function such as light fields with limited number of views without glasses. A view-dependent pixel can be implemented by multiplexing the image pixels of different views and rely on a lenticule arranged in an array or slits arranged in parallel (parallax barriers) to angularly separate (filter) them to create the given color at the desired viewing direction. This is illustrated in Fig. 4 for a display using the lenticular array and parallax barriers, respectively. To suppress the artifacts due to the periodic lenticules and slits in automultiscopic displays, they are slightly slanted so that pixels of a given view are made nonuniform rather than widely separated as periodic vertical lines. Prefiltering has to been applied to generate the nonuniform samples so as to avoid aliasing due to down-sampling since all the pixels from all views have to be multiplexed on the same display [10].

The bandwidth of such displays is also limited because an object with increasing depth becomes smaller, and smaller and hence it will eventually reach the minimum sampling interval and aliasing will occur. To display a light field on such multiview displays, one may also need to bandlimit the plenoptic function [24] as studied in plenoptic sampling [3].

## Application

The plenoptic function serves an important concept for describing visual information in our world. Its sampling analysis also serves as a basic for designing plenoptic cameras and automultiscopic displays for capturing and displaying such high-dimensional function, respectively.

## Open Problems

The efficient capturing and processing of plenoptic function has always been a problem in visual computing and vision research. Due to the multidimensional nature of the plenoptic function and its dependence on the scene geometry, the analysis is very difficult because the function itself may not even be bandlimited. Appropriate optical elements have to be used to reduce the aliasing due to sampling. This makes a general analysis very difficult, though some simple cases such as Lambertian surface, occlusion, and lighting on light field spectrum have been analyzed and found to be useful in practice. Recent advances in computational photography, microelectronics, and processing algorithms have accelerated the development of more sophisticated capturing devices and techniques to improve the rendering quality and reducing the hardware complexity. However, how to achieve high-quality renderings supporting a wide range of view points in large scale environmental modeling remains open.

## References

1. Adelson EH, Bergen JR (1991) The plenoptic function and the elements of early vision. In: Landy M, Movshon JA (eds) Computational models of visual processing. MIT, Cambridge, MA
2. Adelson EH, Wang JYA (1992) Single lens stereo with plenoptic camera. IEEE Trans Pattern Anal Mach Intell 14(2):99–106
3. Chai JX, Tong X, Chan SC, Shum HY (2000) Plenoptic sampling. In: Proceedings of ACM SIGGRAPH, New Orleans, 307–318
4. Chan SC, Ng KT, Gan ZF, Chan KL, Shum HY (2005) The plenoptic videos. IEEE Trans Circuits Syst Video Technol 15(12):1650–1659
5. Chen SE (1995) QuickTime VR – an image-based approach to virtual environment navigation. In: Proceedings of ACM SIGGRAPH, Los Angeles, 29–38
6. Farid H, Simoncelli EP (1998) Range estimation by optical differentiation. J Opt Soc Am A 15(7):1777–1786
7. Fujii T, Kimoto T, Tanimoto M (1996) Ray space coding for 3D visual communication. In: Proceedings of Picture coding Symposium, Melbourne, 447–451
8. Gortler SJ, Grzeszezuk R, Szeliski R, Cohen MF (1996) The lumigraph. In: Proceedings of ACM SIGGRAPH, New Orleans, 43–54
9. Ives HE (1930) Parallax panoramagrams made with a large diameter lens. J Opt Soc Am 20(6):332–342
10. Konrad J, Halle M (2007) 3-D Displays and signal processing. IEEE Signal Process Mag 24(6):97–111
11. Levoy M, Hanrahan P (1996) Light field rendering. In: Proceedings of ACM SIGGRAPH, New Orleans, 31–42
12. Liang CK, Lin TH, Wong BY, Liu C, Chen HH (2008) Programmable aperture photography: multiplexed light field acquisition. ACM Trans Graph 27(3):55:1–55:10
13. Lippmann MG (1908) Epreuves reversible donnant la sensation du relief. J Phys 7:821–825
14. McMillan L, Bishop G (1995) Plenoptic modeling: an image-based rendering system. In: Proceedings of ACM SIGGRAPH, Los Angeles, 39–46
15. Naemura T, Tago J, Harashima H (2002) Real-time video-based modeling and rendering of 3D scenes. IEEE Comput Graph Appl 22(2):66–73
16. Ng R, Levoy M, Bredif M, Duval G, Harowitz M, Hanrahan P (2005) Light field photography with a hand-held plenoptic camera. Stanford University Computer Science Tech Report CSTR 2005–02. http://graphics.stanford.edu/papers/lfcamera/
17. Shum HY, He LW (1999) Rendering with concentric mosaics. In: Proceedings of ACM SIGGRAPH, Los Angeles, 299–306
18. Shum HY, Chan SC, Kang SB (2007) Image-based rendering. Springer, New York
19. Szeliski R, Shum HY (1997) Creating full view panoramic image mosaics and environment maps. In: Proceedings of ACM SIGGRAPH, Los Angeles, 251–258
20. Wong T, Fu C, Heng P, Leung C (2002) The plenoptic illumination function. IEEE Trans Multimed 4(3):361–371
21. Wilburn B, Joshi N, Vaish V, Talvala E, Antunez E, Barth A, Adams A, Horowitz M, Levoy M (2005) High performance imaging using large camera arrays. ACM Trans Graph 24(3):765–776
22. Yang JC, Everett M, Buehler C, McMillan L (2002) A real-time distributed light field camera. In: Proceedings of Eurographics Workshop on Rendering, Pisa, 77–86
23. Zitnick CL, Kang SB, Uyttendaele M, Winder S, Szeliski R (2004) High-quality video view interpolation using a layered representation. In: Proceedings of ACM SIGGRAPH, Los Angeles, 600–608
24. Zwicker M, Vetro A, Yea S, Matusik W, Pfister H, Durand F (2007) Resampling, antialiasing, and compression in multiview 3-D display. IEEE Signal Process Mag 24(6):88–96

## Point Spread Function Estimation

►Blur Estimation

# Polarization

Gary A. Atkinson
Machine Vision Laboratory, University of the West of England, Bristol, UK

## Related Concepts

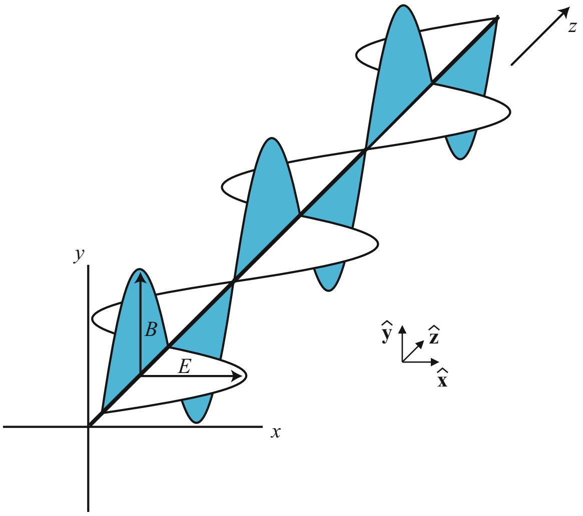
▸Fresnel Equations; ▸Polarized Light in Computer Vision; ▸Polarizer

## Definition

Polarization refers to the orientation distribution of the electromagnetic waves that constitute light rays. Light can assume a range of polarization states including unpolarized (uniformly mixed orientations), linearly polarized (fixed orientations), and circularly polarized (rotating field about the line of sight).

## Background

The polarization of light is a property that relates to the directionality of the constituent electric and magnetic fields. In free space, light consists of sinusoidally oscillating electric and magnetic fields that are orthogonal to each other and to the direction of propagation. In the case of *perfectly linearly polarized light*, as shown in Fig. 1, the two types of field remain in a fixed plane through space. For *unpolarized light*, the orientations of these planes are randomized through temporal and spatial location. *Circular and elliptical polarizations* are also possible, whereby in a plane normal to the ray, the field is radial at each point with an orientation following a circular or elliptical temporal pattern. Finally, it is possible to have light consisting of combinations of polarization types. For example, *partially linearly polarized light* involves fields oscillating in all directions, but with one particular preferred direction. Polarization can be caused by a range of phenomena such as scattering, reflection, and transmission through media interfaces. In most computer vision and industrial applications, polarized light is generated and measured using polarizing filters.



**Polarization, Fig. 1** A linearly polarized wave consisting of vertical magnetic fields ($B$) and horizontal electric fields ($E$)

## Theory

### Plane Wave Versus Photon Representation

Most texts on polarization regard light as an electromagnetic wave and explain the various phenomena via interactions of the constituent electric and magnetic fields with matter or other waves. However, in the realm of quantum optics, light consists of streams of particles known as photons. The details of this [1] are beyond the scope of this entry, although it is worthwhile noting a few basic points:

- Each photon has an associated energy that is proportional to its frequency.
- The angular momentum of each individual photon is quantized and can only take values of $-h/2\pi$ or $+h/2\pi$, where $h$ is the *Planck Constant* [2].
- Circular polarization consists of identical photons (for monochromatic light) all with angular momentum vectors oriented either parallel to the direction of propagation (left- or $\mathscr{L}$-state) or antiparallel to the direction of propagation (right- or $\mathscr{R}$-state).
- Linear polarization ($\mathscr{P}$-state) can be described by a linear combination of $\mathscr{L}$- and $\mathscr{R}$-states.

For the remainder of this entry, the electromagnetic wave description of light will be employed.

### Linear Polarization

In linearly polarized light, the electric and magnetic fields assume a fixed orientation through space, as shown in Fig. 1. For the remainder of this entry, the focus will be on the electric field (similar arguments to those below can be applied to the magnetic component – see [3] for a thorough description). In the simplest case, the electric wave of linearly polarized light can be described using a standard equation of a plane wave:

$$\mathbf{E}(z,t) = \hat{\mathbf{x}}E_0 \cos(kz - \omega t) \tag{1}$$

where the coordinate axes $(x, y, z)$ and unit vectors $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, and $\hat{\mathbf{z}}$ are defined in Fig. 1, $E_0$ is the amplitude of the wave, $k$ is the wave number, $\omega$ is the angular frequency, and $t$ is time. A detailed overview of the construction of the plane wave equation and its manipulation can be found in [4].

The wave described by (Eq. 1) is horizontally oriented. Of course, it is possible to replace the $\hat{\mathbf{x}}$ in (Eq. 1) with $\hat{\mathbf{y}}$, in which case, a vertical plane polarized wave will be represented. In general, an arbitrary case can be represented as a sum of orthogonal components:

$$\mathbf{E}(z,t) = \mathbf{E}_x(z,t) + \mathbf{E}_y(z,t) \tag{2}$$

where

$$\begin{aligned}\mathbf{E}_x(z,t) &= \hat{\mathbf{x}}E_{0x}\cos(kz - \omega t)\\ \mathbf{E}_y(z,t) &= \hat{\mathbf{y}}E_{0y}\cos(kz - \omega t + \phi)\end{aligned} \tag{3}$$

Here, $E_{0x}$ and $E_{0y}$ refer to the amplitude components in the $x$ and $y$ directions, respectively, and $\phi$ is the *phase difference* between the components (the absolute phases are irrelevant).

A wave is linearly polarized wherever $\phi = m\pi$, where $m$ is an integer:

$$\begin{aligned}\mathbf{E}(z,t) &= \left(\hat{\mathbf{x}}E_{0x} + \hat{\mathbf{y}}E_{0y}\right)\cos(kz - \omega t) \quad \text{even } m\\ \mathbf{E}(z,t) &= \left(\hat{\mathbf{x}}E_{0x} - \hat{\mathbf{y}}E_{0y}\right)\cos(kz - \omega t) \quad \text{odd } m\end{aligned} \tag{4}$$

where $\left(\hat{\mathbf{x}}E_{0x} + \hat{\mathbf{y}}E_{0y}\right)$ or $\left(\hat{\mathbf{x}}E_{0x} - \hat{\mathbf{y}}E_{0y}\right)$ define both the amplitude and direction of the electric field (or *plane of polarization*) for each case.

## Circular and Elliptical Polarization

The linear polarization case of (Eq. 4) is essentially a special case of the general electric field equation (Eq. 2). A second special case is where $\phi = \pi(m - 1/2)$, where $m$ is, again, an integer. For the case where $E_{0x} = E_{0y} = E_0$, (Eq. 3) can be substituted into (Eq. 2), to generate the following wave:

$$\begin{aligned}\mathbf{E}(z,t) &= E_0\left(\hat{\mathbf{x}}\cos(kz - \omega t) + \hat{\mathbf{y}}\sin(kz - \omega t)\right) \text{ even } m\\ \mathbf{E}(z,t) &= E_0\left(\hat{\mathbf{x}}\cos(kz - \omega t) - \hat{\mathbf{y}}\sin(kz - \omega t)\right) \text{ odd } m\end{aligned} \tag{5}$$

This case has two interesting properties. Firstly, in contrast to the linearly polarized case (Eq. 4), the scalar amplitude is fixed at $E_0$. Secondly, the direction of the electric field is rotating at a fixed angular velocity in a clockwise (for even $m$) or anticlockwise (for odd $m$) direction relative to the direction of propagation. An alternative way to envisage this is that for a given point along the line of propagation of the wave, the tip of the electric field vector follows a circular motion about that point. The clockwise case is referred to in the optics literature as *right circularly polarized* light, while the anticlockwise case is called *left circularly polarized*. Note that a superposition of right and left circularly polarized light gives rise to a linearly polarized wave of amplitude $2E_0$. This is conducive to the photon angular momentum arguments discussed earlier.

For the general case, where $E_{0x} \neq E_{0y}$ and for nonintegral values of $m$, the tip of the electric field vector follows an elliptical path. This can be found by seeking a time- and position-independent solution to (Eq. 2) and (Eq. 3). The result [1] is that

$$\left(\frac{E_x}{E_{0x}}\right)^2 + \left(\frac{E_y}{E_{0y}}\right)^2 - 2\left(\frac{E_x}{E_{0x}}\right)\left(\frac{E_y}{E_{0y}}\right)\cos\phi = \sin^2\phi \tag{6}$$

This is the equation of an ellipse oriented at an angle, $\gamma$, given by

$$\tan 2\gamma = \frac{2E_{0x}E_{0y}\cos\phi}{E_{0x}^2 - E_{0y}^2} \tag{7}$$

For the simplest case, where $\gamma = 0$ the relationship reduces to

$$\frac{E_x^2}{E_{0x}^2} + \frac{E_y^2}{E_{0y}^2} = 1 \tag{8}$$

Elliptically polarized light is often referred to as the $\mathscr{E}$-state.

## Partial Linear Polarization

Often in nature and technology, light assumes a partially polarized form. The most common of these is partially linearly polarized light. This consists of a superposition of linearly polarized light and unpolarized light. It is often desirable to define a quantity known as the *degree of polarization*, $\rho$, that relates the intensity (or flux density) of the polarized component, $I_p$, to that of the unpolarized component, $I_u$:

$$\rho = \frac{I_p}{I_p + I_u} \tag{9}$$

This is often expressed as a percentage, such that light of equal components is 50% polarized. The degree of polarization can be analyzed using a polarizing filter (often referred to as an "analyzer" in optics texts). If the maximum and minimum transmitted light intensities as the polarizer is rotated are $I_{\max}$ and $I_{\min}$, then $I_{\min} = I_u/2$ and $I_{\max} = I_u/2 + I_p$ so that

$$\rho = \frac{I_{\max} - I_{\min}}{I_{\max} + I_{\min}} \tag{10}$$

## Alternative Representation

### Stokes Vectors

In many practical applications of polarized light, the physical wave description above is of little use due to the minuscule quantities involved. The Stokes vectors aim to represent the polarization state of light in a compact form that most detectors can measure (See [5] for derivations and further details).

The general Stokes vector is defined by

$$S = [S_0, S_1, S_2, S_3]^T \tag{11}$$

The first of the parameters of the Stokes vector, $S_0$, is simply the intensity of the light. The second parameter, $S_1$, quantifies the tendency for vertical or horizontal polarization and is related to the difference in the corresponding vertical and horizontal components of the wave's electric field. Where $S_1 > 0$, the light resembles a horizontal $\mathscr{P}$-state, while $S_1 < 0$ relates to vertical $\mathscr{P}$-states. For the case where $S_1 = 0$, there is no tendency either way as in circular polarized light, elliptical light at 45° and unpolarized light. $S_2$ is interpreted in a similar fashion, but relates to angles of 45° and $-45°$. Finally, $S_3$ relates to the handedness of the polarization ($S_3 > 0$ is right-handed, $S_3 < 0$ is left-handed and $S_3 = 0$ is neither).

Often, the Stokes vectors are normalized by $S_0$ so that, for example, unpolarized light has $S = [1, 0, 0, 0]^T$; vertically linearly polarized light has $S = [1, -1, 0, 0]^T$; linearly polarized light at 45° has $S = [1, 0, 1, 0]^T$; and right-circularly polarized light has $S = [1, 0, 0, 1]^T$. Furthermore, light may consist of superpositions of different states, in which case the individual vectors are simply added together.

The Stokes vectors provide useful representations of polarization for *incoherent light*, which is abundant in nature. Computer vision and other fields may utilize the Stokes representation of the degree of polarization. In the case of partially linearly polarized light, for example, the degree of polarization is

$$\rho = \frac{\sqrt{S_1^2 + S_2^2}}{S_0} \tag{12}$$

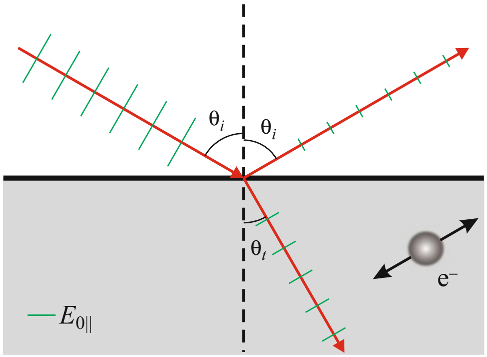
### Jones Vectors

For *coherent light*, typically from lasers, an alternate representation is also possible. The Jones vector of a polarized light wave is simply given by

$$\mathbf{E} = \begin{bmatrix} E_x(t) \\ E_y(t) \end{bmatrix} \tag{13}$$

The electric fields in the Jones vectors are generally represented as a wave via the imaginary number approach [4] ($E = E_0 e^{i(kz - \omega t + \phi)}$). They are then normalized by the intensity in a similar fashion to the Stokes vectors. The results [1] are that horizontal and vertical $\mathscr{P}$-states are given by $[1, 0]^T$ and $[0, 1]^T$, respectively; $\mathscr{P}$-states at 45° are given by $\left(1/\sqrt{2}\right)[1, 1]^T$ and $\left(1/\sqrt{2}\right)[1, -1]^T$; and $\mathscr{R}$- and $\mathscr{L}$-states are given by $\left(1/\sqrt{2}\right)[1, -i]^T$ and $\left(1/\sqrt{2}\right)[1, i]^T$.

The advantage of the Jones representation (aside from its compactness) is that certain optical operations can be described sequentially in terms of matrix operations. For example, light passing through a linear vertical polarizer has its Jones vector modified by the a *Jones matrix* of $\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$. A related set of matrices for operating on Stokes vectors has also been derived. These are called *Mueller matrices* and have dimensions of $4 \times 4$. Further details of Jones and Mueller matrices can be found in [1] and references therein.



**Polarization, Fig. 2** Polarization from specular reflection

## Polarized Light in Nature

Sunlight, which constitutes the overwhelming majority of natural light on Earth, is unpolarized. Despite this, linearly polarized light is abundant in nature as a result of various phenomena that convert unpolarized light to polarized light. Circularly polarized light by contrast is rare in nature and only occurs under certain ideal conditions. The two most common causes of polarization are reflection from surfaces and light scattering in the atmosphere and these are discussed below. Other causes [6] include internal reflection, rainbows, clouds, and birefringence (in materials where optical properties are anisotropic).

### Reflection

Consider a light wave impinging upon a smooth dielectric surface, as shown in Fig. 2. A fraction of the light penetrates the surface and the remainder is reflected. In real surfaces, the penetrated light undergoes a series of complicated scattering interactions according to the radiative transfer equation [7]. However, the problem can be simplified by assuming that no scattering occurs apart from at the interface. In this case, the transmitted wave is refracted according to the well-known Snell's Law [1]:

$$n_i \sin\theta_i = n_t \sin\theta_t \tag{14}$$

where $n_i$ and $n_t$ refer to the refractive indices of air ($\approx$1) and the reflecting material and $\theta_i$ and $\theta_t$ refer to the incident and transmitted angles respectively.

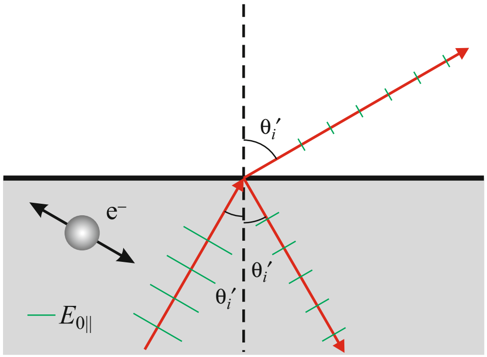
Consider the component of the electric field of the wave that is parallel to the plane of incidence. This component results in electron dipole oscillations in the medium at angle $\theta_t$ (see Fig. 2). These dipoles close to the interface emit the reflected wave. However, this reflected wave is at a different angle to $\theta_t$ meaning that the parallel component of the reflected wave is attenuated due to foreshortening. For the component of the electric field that is perpendicular to the plane of incidence, the foreshortening effect is not present. This difference in wave attenuation caused by foreshortening gives rise to a partially linearly polarized reflection.

Note that, for the case where $\theta_i + \theta_r = 90°$, foreshortening eliminates the parallel component of the reflection entirely. The angle at which this occurs is known as the *Brewster Angle*, $\theta_B$, and it follows from (Eq. 14) that, for the case where $n_i = 1$,

$$\theta_B = \arctan n_t \tag{15}$$

Much of the light present in everyday life is not specularly reflected from surfaces, as discussed above, but diffusely reflected. That is, the light is scattered either at the surface or below the surface and both of these phenomena affect the polarization state of the reflected light. For the latter case, several efforts have been made to model the precise means by which light is scattered to form a reflection [8]. The important point, so far as polarization is concerned, is that after undergoing internal scattering, the light impinges upon the air-medium interface from within, as shown in Fig. 3. This light then undergoes a refraction as it passes the interface. A similar argument to that above for specular reflection can then be applied to establish that a foreshortening effect polarizes the diffusely reflected light also. Note however, that as the incident wave can never be perpendicular to the reflected wave, there is no angle by which the diffusely reflected light becomes completely polarized. The degree and angles of polarization can be determined from the Fresnel equations and are discussed in detail in the entry ▶Polarized Light in Computer Vision. Note also that polarization by refraction occurs as light from the Sun enters water, causing light near the surface of lakes, seas, and oceans to be partially polarized.

For the case of surface scattering, several attempts to model the microscopic roughness patterns have been made, such as [9] and [10]. For both specular and diffuse reflection, the random nature of microscopic surface corrugations has the effect of mixing the orientations of reflected light fields. One of the effects



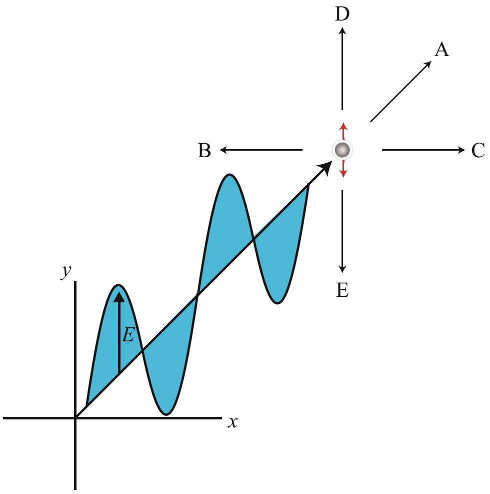
**Polarization, Fig. 3** Polarization from diffuse reflection (refraction)

of this is to depolarize the reflection. As predicted by the *Umov Effect* [11], rough surfaces with lower albedo retain higher degrees of polarization than light surfaces. This is due to the fact that fewer inter-reflections take place on dark surfaces before the light waves are completely absorbed by the medium. This effect was first noted in astronomy, where the degree of polarization of an astronomical body is inversely proportional to its albedo.

### Scattering

As unpolarized light is scattered by small particles (for e.g., a molecule in the atmosphere or a dust particle) it becomes partially linearly polarized. The effect can be explained in a similar fashion to that with surface reflection. Consider a light wave impinging on a particle. The particle will scatter the light in a range of directions. It turns out that, when viewed at an angle of 90° to the direction of the light wave, the scattered light is completely polarized, while the waves that are scattered close to the direction of propagation or scattered by ≈180° are only slightly polarized. This phenomenon is observed on a grand scale in the Earth's atmosphere. When a clear sky is viewed at an angle of 90° from the Sun, it can be observed to be ≈75% polarized, compared to almost no polarization at ≈0° and appraching 180°. In practice, light scattered at 90° is not completely polarized due to a range of factors such as molecular anisotropies and multiple-scattering depolarization.

The phenomenon of polarization by scattering can be explained as follows. The wave shown in Fig. 4 induces an oscillating dipole in the scattering particle as a result of the redistribution of the electron cloud and nucleus location caused by the electric field. The dipole, in turn, generates the scattered wave. Assume that the impinging wave is linearly vertically polarized, as in Fig. 4. The new wave has maximum amplitude at an angle perpendicular to the dipole orientation and zero amplitude when parallel to the dipole due to foreshortening. In relation to Fig. 4, amplitude maxima would be found in directions $A$, $B$, and $C$ (in addition to backscatter), while no scattering occurs in directions $D$ or $E$. Next consider an impinging wave that is polarized horizontally. In this case, directions $A$, $D$, and $E$ are the maxima, while directions $B$ and $C$ have no scattering. Finally, by representing an unpolarized incident wave as a linear combination of randomized linearly polarized states, it can be seen that direction $A$ and backscatter remain unpolarized, directions $B$ and $C$ retain only the vertical polarization and directions $D$ and $E$ retain only the horizontal polarization. Directions intermediate between those discussed will, of course, become partially linearly polarized. For such angles with the common case of scattering by atmospheric dust or smoke, the degree of polarization is given by [3]



**Polarization, Fig. 4** Polarization from scattering

$$\rho = \frac{\sin^2 \beta}{1 + \cos^2 \beta} \tag{16}$$

where $\beta$ is the scattering angle. This is known as Rayleigh scattering. However, complete polarization is not typically observed due to the reasons mentioned above.

## Application

Many creatures, including some insects and marine animals, have eyes that are sensitive to the polarization state of light [12]. Bees and ants, for example, use the polarization pattern in the sky or water to aid navigation [6]. Several marine creatures use similar patterns found underwater for the same purpose [13]. There is evidence [14] that certain aquatic creatures use polarization to isolate objects of interest within their field of view. It is also believed that some species communicate by reflecting light only of a certain polarization angle.

In everyday life, a common application of polarization is in sunglasses and camera filters. A pair of polarizing sunglasses with vertical transmission axis blocks out large amounts of specular glare from surfaces as Fig. 2 suggests. A correctly oriented camera lens filter can also be used to diminish glare. In addition, such a filter can increase contrast between clouds and the sky by cutting out the polarized component of sky and having less impact on light from clouds.

Liquid crystal displays (LCDs) are based on crystals placed between crossed-polarizers. Light is transmitted or absorbed in different segments, depending on which areas is subjected to a voltage. Where the voltage is applied, the crystals rotate the angle of polarization by 90°. Stress patterns of many materials can be analyzed by placing samples between crossed polarizers and measuring the transmitted intensity patterns. This is possible due to the changing birefringence properties of materials under strain. In entertainment, 3D images can be generated by time-multiplexing different polarization states onto a screen, with each state representing a different viewpoint of the object being displayed. The viewers wear polarizing spectacles that transmit each state to separate eyes. Finally, polarization is prevalent in astronomy. For example, transmission and absorption effects in strongly magnetic regions of the Sun's surface cause circular polarization, while interstellar dust is responsible for polarization by scattering over long distances.

Within the field of computer vision, polarization has been used for a range of tasks, including the following (further details and references can be found in the separate entry on ►Polarized Light in Computer Vision):

- Specularity reduction
- Shape recovery
- Reflectance analysis
- Image enhancement
- Reduction of inter-reflections
- Separation of reflectance components
- Image segmentation

## References

1. Hecht E (1998) Optics, 3rd edn. Addison Wesley, Reading
2. Tipler PA, Mosca G (2007) Physics for scientists and engineers, 6th edn. Freeman, New York
3. Born M, Wolf E (1999) Principles of optics. Electromagnetic theory of propagation, interference and diffraction of light, 7th edn. Pergamon, London
4. Main IG (1993) Vibrations and waves in physics, 3rd edn. Cambridge University Press, Cambridge
5. Shurcliff WA (1962) Polarized light: production and use. Harvard University Press, Cambridge
6. Können GP (1985) Polarized light in nature. Cambridge University Press, Cambridge
7. Chandrasekhar S (1960) Radiative transfer. Oxford University Press, New York
8. Jensen HW, Marschner SR, Levoy M, Hanrahan P (2001) A practical model for subsurface light transport. In: Proceedings of SIGGRAPH, Los Angeles, pp 511–519
9. Torrance K, Sparrow M (1967) Theory for off-specular reflection from roughened surfaces. J Opt Soc Am 57: 1105–1114
10. Beckmann P, Spizzichino A (1963) The scattering of electromagnetic waves from rough surfaces. Pergamon, New York
11. Umov N (1905) Chromatische depolarisation durch lichtzerstreuung. Phys Z 6:674–676
12. Shashar N, Cronin TW, Wolff LB, Condon MA (1998) The polarization of light in a tropical rain forest. Biotropica 30:275–285
13. Horváth G, Varjú D (1995) Underwater polarization patterns of skylight perceived by aquatic animals through Snell's window on the flat water surface. Vision Res 35: 1651–1666
14. Cronin TW, Shashar N, Caldwell RL, Marshall J, Cheroske AG, Chiou TH (2003) Polarization and its role in biological signalling. Integr Comp Biol 43:549–558

## Polarization Filter

►Polarizer

# Polarized Light in Computer Vision

Gary A. Atkinson
Machine Vision Laboratory, University of the West of England, Bristol, UK

## Related Concepts

▶Fresnel Equations; ▶Polarization; ▶Polarizer; ▶Transparent Layers

## Definition

Polarization refers to the orientation distribution of the electromagnetic waves that constitute light rays. The phenomenon of light polarization has been exploited in computer vision for a range of applications including surface reconstruction, specular/diffuse separation, and image enhancement.

## Background

Light consists of orthogonal electric and magnetic fields. Most natural light is unpolarized and so consists of randomly fluctuating field directions. However, a range of natural phenomena (e.g., scattering and reflection) and human inventions (e.g., polarizing filters and liquid crystal displays) cause the light to become polarized. That is, the electric and magnetic fields become confined to specific planes or get constrained in other ways. In the field of computer vision, both natural and artificially generated polarized light has been utilized for a range of applications including specularity reduction, shape recovery, reflectance analysis, image enhancement, segmentation, and separation of reflectance components.

## Theory

The majority of research into polarization methods for computer vision rely on passively analyzing the polarization state of incoming light using either uncontrolled, or at least unpolarized, illumination. However, one of the most common uses of polarization in computer vision is to remove specularities from images using a more active approach. This entry first describes this method and is followed by descriptions of techniques for polarization measurement. It then covers the relevant theory of the two most common natural causes of polarization (reflection and scattering) and their exploitation in computer vision.

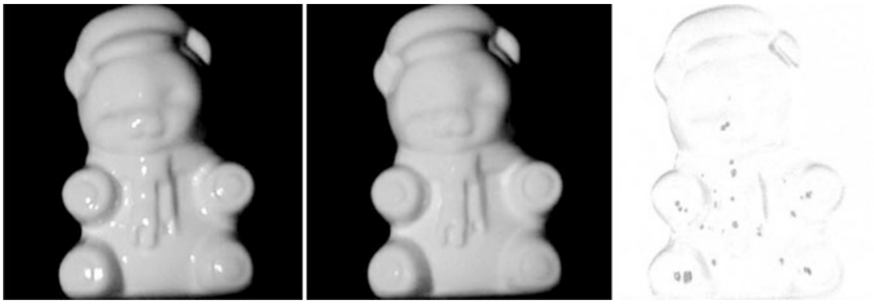
### Polarization and Specularities

The images of the porcelain bear model in Fig. 1 were obtained using a camera with one linear polarizer (often referred to as an "analyzer" in optics texts) mounted on the lens and another in front of the only light source. For the first image, the two polarizers were oriented parallel to each other, while for the second image they were at 90°. The second image clearly shows a minimization of specularity. This is of great benefit to a range of computer vision methods that assume specularities are absent from images such as shape-from-shading [1] and photometric stereo [2].

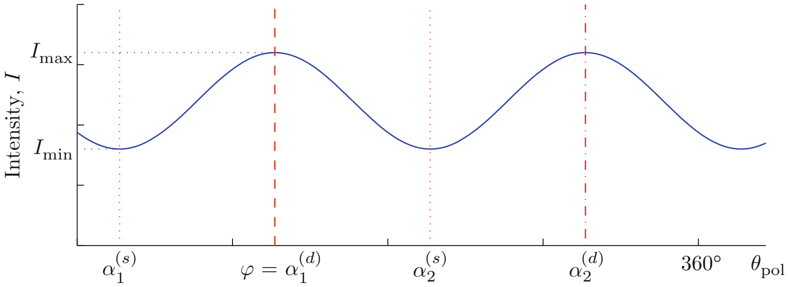
The theory behind the specularity reduction is that the polarized incoming light induces electron oscillations in the reflecting medium in only one direction (see the entry on ▶Polarization). This means that the *specularly* reflected light, which is generated by these oscillations, has an electric field that is constrained to a single plane also. If the lens polarizer is oriented at right angles to this plane then all of this light is blocked from the camera. On the other hand, the *diffusely* reflected light is depolarized by subsurface scattering and surface roughness. This means that much of the diffusely reflected light is transmitted through the lens polarizer.

### Polarization Imaging

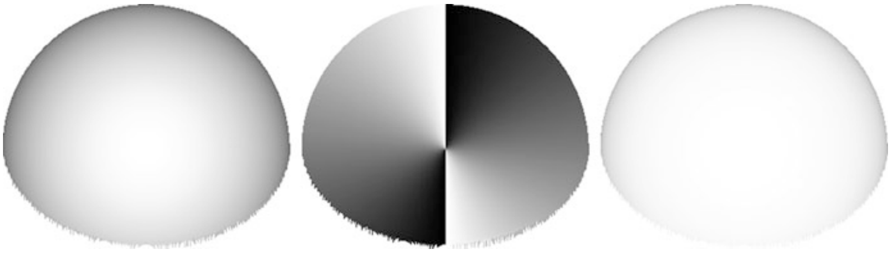
For the cases where the incident light is unpolarized, the simplest and most common method of measuring polarization is to take a sequence of images with a rotating linear polarizer. Consider a camera's field of view where the incoming light is partially linearly polarized with varying degrees between 0% and 100% (see the entry on ▶Polarization for the definition of the degree of polarization). The aim is to acquire a *polarization image*: a three-channel image where each pixel has components corresponding to (1) the grayscale intensity, (2) the degree of polarization, and (3) the phase angle (not to be confused with the phase of the



**Polarized Light in Computer Vision, Fig. 1** *Left*: Uncrossed polarizers, *center*: crossed polarizers, *right*: difference image [3]. N.B. The difference image has been inverted and undergone nonlinear intensity scaling to aid clarity



**Polarized Light in Computer Vision, Fig. 2** Transmitted radiance sinusoid. $\alpha_1^{(s)}$ and $\alpha_2^{(s)}$ are the two possible surface azimuth angles for a given phase angle $\varphi$, assuming that the reflection is specular. For diffuse reflection, the possible azimuth angles are $\alpha_1^{(d)}$ and $\alpha_2^{(d)}$



**Polarized Light in Computer Vision, Fig. 3** Synthetic rendering of a polarization image of a hemisphere. *Left*: intensity, *center*: phase angle, *right*: degree of polarization

electric and magnetic fields of the wave that constitute the light ray) of the linearly polarized component of the incoming light.

As the polarizer on the lens rotates, the transmitted intensity at each point is given by the *transmitted radiance sinusoid*:

$$I\left(\theta_{\text{pol}}, \varphi\right) = \frac{I_{\max} + I_{\min}}{2} + \frac{I_{\max} - I_{\min}}{2} \cos\left(2\theta_{\text{pol}} - 2\varphi\right) \quad (1)$$

where $\theta_{\text{pol}}$ is the polarizer orientation, $I_{\min}$ and $I_{\max}$ are the minimum and maximum observed pixel brightnesses and $\varphi$ is the phase angle. The transmitted radiance sinusoid is shown graphically in Fig. 2.

The complete polarization image is then constructed by fitting the transmitted radiance sinusoid to three or more different intensity images for different polarizer angles. This assumes that the pixel brightness is proportional to the light impinging on the sensor – i.e., the camera response function is linear. For typical imaging equipment, the intensity component is in the range $[0, 255]$, the degree of polarization is always in the range $[0, 1]$, and the phase is always in the range $[0, 180°)$. Fig. 3 shows a synthetic example of the three components of a polarization image.

A major weakness of many polarization methods is the amount of time needed to acquire the data (the actual processing of the data is often highly efficient). For the rotating polarizer method described above, three or more images are required with the polarizer at different orientations. This limits applications to static scenes. Matters were improved after the development of *polarization cameras* [4] that used liquid crystals to rapidly switch the axis of the polarizing filter. Subsequently, PLZT (polarized lead zirconium titanate) cameras [5] were designed with the aim of recovering all four components of the Stokes vectors (defined in the entry on ▶Polarization) [6].

Other methods of rapid polarization image capture include placing multiple cameras with different polarizers but near-parallel optical axes [7]; using a detector array-like Bayer pattern to capture the polarization field on chip [8]; using a lightfield camera, where the aperture is divided into segments, some of which contain polarizers [9]; and employing multiple CCDs to capture all the polarization components using beam splitters [10]. The interested reader is also referred to [11] for a detailed survey.

## Polarization and Reflection in Images

Perhaps the most common exploitation of polarization in computer vision is to measure the polarization state of light reflected from dielectric surfaces [12]. As described in the entry on ▶Polarization, natural light undergoes partial polarization when it is reflected from surfaces. Consider a specular reflection being viewed through a rotating polarizer. Because the electric field of the reflected wave is most attenuated in the plane of reflection, the greatest transmission through the polarizer occurs when it is oriented at right angles to the plane. Minimum transmission occurs when the polarizer is parallel to the plane. Note however, that the polarizer is unable to distinguish two planes of reflection oriented 180° apart. If $\alpha$ is the azimuth angle of the surface (the angle of the projection of the surface normal onto the image plane), then there are two possible values for a given measurement of $\varphi$:

$$\alpha_1^{(s)} = \varphi - 90° \qquad \alpha_2^{(s)} = \varphi + 90° \tag{2}$$

where the superscript $s$ indicates that the reflection is specular.

As explained in [12], the Fresnel equations can be used to represent the maximum and minimum intensities passing through a rotating polarizer as

$$\begin{aligned} I_{\min} &= \frac{R_{\parallel}(n,\theta)}{R_{\perp}(n,\theta) + R_{\parallel}(n,\theta)} I_r \\ I_{\max} &= \frac{R_{\perp}(n,\theta)}{R_{\perp}(n,\theta) + R_{\parallel}(n,\theta)} I_r \end{aligned} \tag{3}$$

Here $R_{\parallel}$ and $R_{\perp}$ are the parallel and perpendicular Fresnel intensity reflectivity components, respectively (see the entry on ▶Fresnel Equations), $n$ is the refractive index, $\theta$ is the angle of incidence, and $I_r$ is the magnitude of the specularity. In an imaging situation, $\theta$ is equivalent to the zenith angle of the surface (the angle between the surface normal and the line of sight of the camera). Rewriting in terms of the degree of polarization and rearranging gives [12]

$$\rho_s(n,\theta) = \frac{2\sin^2\theta\cos\theta\sqrt{n^2 - \sin^2\theta}}{n^2 - \sin^2\theta - n^2\sin^2\theta + 2\sin^4\theta} \tag{4}$$

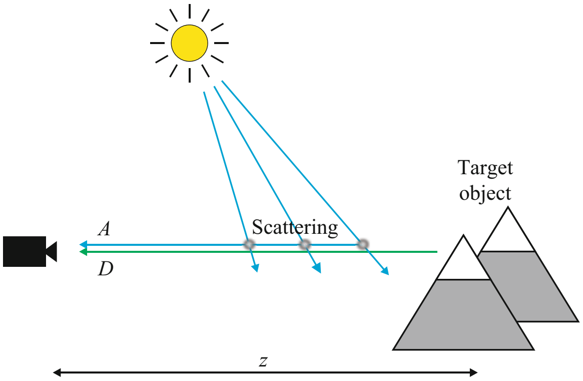
where the suffix $s$ is appended to the degree of polarization as this assumes a purely specular reflection.

Equations 2 and Eq. 4 allow one to impose constraints on the angle of a reflecting surface for shape recovery applications using measurements taken of $\varphi$ and $\rho$. The azimuth angle of the surface has two possibilities as shown by (Eq. 2) and Fig. 2. Apart from at the Brewster angle, where $\rho = 1$, there are also two solutions to (Eq. 4), meaning two zenith angles are also possible.

Combining the above theory with the subsurface scattering model of reflection [12, 13], the following relations for *diffuse* reflection can be derived:

$$\alpha_1^{(d)} = \varphi \qquad \alpha_2^{(d)} = \varphi + 180° \tag{5}$$

$$\rho_d = \frac{(n - 1/n)^2\sin^2\theta}{2 + 2n^2 - (n + 1/n)^2\sin^2\theta + 4\cos\theta\sqrt{n^2 - \sin^2\theta}} \tag{6}$$



**Polarized Light in Computer Vision, Fig. 4** Haze in an image due to atmospheric scattering

On this occasion, there is only one solution for the zenith angle, but the degree of polarization is lower and therefore more difficult to measure. Indeed, the information contained in this degree of polarization is only useful for large zenith angles [14].

### Outdoor Polarization Imaging

One of the most common causes of polarization in nature is atmospheric scattering [15]. Consider the imaging process of a distant outdoor object or scene. Light that arrives at a detector consists of a component transmitted from the object itself, $D$, and a component of "airlight," $A$. The former of these is typically reflected sunlight while the latter is light scattered from the atmosphere, as shown in Fig. 4. The total intensity at the camera can be represented by

$$I_{\text{tot}} = A + D \tag{7}$$

It can be shown [16] that

$$A = A_\infty (1 - t(z)) \tag{8}$$

$$D = L_{\text{obj}} t(z) \tag{9}$$

where $A_\infty$ is the airlight radiance for an object at infinity, $L_{\text{obj}}$ is the object radiance, $z$ is the distance to the object and $t(z)$ is given by

$$t(z) = \exp\left(-\int_0^z \beta(z')\mathrm{d}z'\right) \tag{10}$$

where $\beta$ is called the *coefficient of extinction*.

As explained in the ▸Polarization entry, atmospheric scattering causes polarization. The degree of polarization for the airlight component is be given by

$$\rho = \frac{A_\perp - A_\parallel}{A_\perp + A_\parallel} \tag{11}$$

where $A_\perp$ and $A_\parallel$ are the scattered components parallel to and perpendicular to the plane of scattering, respectively. If it is assumed that the target object is emitting/reflecting unpolarized light, then the light measured through a linear polarizer is a transmitted radiance sinusoid, as in Fig. 2, but with an intensity offset equal to the light directly transmitted from the target object. Clearly, the simple task of rotating a lens-mounted polarizer to the angle that allows minimum transmission removes some of the airlight and so enhances the image of the target object. Taking images at more than one polarizer orientation allows to solve for the unknowns in (Eq. 7)–(Eq. 11) and dehaze an image more effectively [16]. A related polarization analysis can also be used for images taken in murky water [17].

## Application

The above theory, and extensions thereof, has been used in a range of areas within computer vision. Perhaps the most studied is in shape recovery. This is due to the ease by which strong constraints can be placed on the surface normals from single viewpoints. Polarization therefore offers huge benefits to shape-from-shading and related methods. As explained above, the normals are not fully constrained using these techniques so additional methods are required to complete the reconstruction [14, 18]. It is also possible to model surfaces that are usually inaccessible to computer vision such as transparent surfaces [18]. As already mentioned, polarization from the atmosphere has been applied to image enhancement outdoors [16] and underwater [17].

The difference in polarizing properties between metallic and dielectric reflection [15] has been used for segmentation [12], while the difference between specular and diffuse properties has allowed the two reflection components to be separated [19]. Related work has allowed improved laser-based range finding, by minimizing the effects of inter-reflections [20]. Other work has exploited polarization by specular reflection for separating transmitted scenes from reflected scenes in a glass window [21, 22]. Polarization has also found a range of other applications including reflectance analysis [23] and cosmetics [24].

## References

1. Horn BKP, Brooks MJ (1989)
   Shape from shading. MIT, Cambridge
2. Woodham RJ (1980) Photometric method for determining surface orientation from multiple images. Opt Eng 19: 139–144
3. Ragheb H, Hancock ER (2002) Highlight removal using shape-from-shading. In: Proceedings of European conference on computer vision (ECCV). Springer, Berlin/New York, pp 626–641
4. Wolff LB (1997) Polarization vision: a new sensory approach to image understanding. Image Vis Comput 15:81–93
5. Shames PE, Sun PC, Fainman Y (1998) Modelling of scattering and depolarizing electro-optic devices. I. characterization of lanthanum-modified lead zirconate titanate. Appl Opt 37:3717–3725
6. Miyazaki D, Takashima N, Yoshida A, Harashima E, Ikeuchi K (2005) Polarization-based shape estimation of transparent objects by using raytracing and PLZT camera. Proc SPIE 5888:1–14
7. Hooper BA, Baxter B, Piotrowski C, Williams JZ, Dugan J (2009) An airborne imaging multispectral polarimeter. In: Proceedings of IEEE/MTS Oceans, Biloxi, Mississippi, USA
8. Gruev V, der Spiegel JV, Enghet N (2009) Advances in integrated polarization image sensors. In: Proceedings of LiSSA Workshop, Bethesda, Maryland, USA, pp 62–65
9. Horstmeyer R, Euliss G, Athale R, Levoy M (2009) Flexible multimodal camera using a light field architecture. In: Proceedings of computational photography (ICCP). IEEE, Piscataway, pp 1–8
10. Zappa CJ, Banner ML, Schultz H, Corrada-Emmanuel A, Wolff LB, Yalcin J (2008) Retrieval of short ocean wave slope using polarimetric imaging. Meas Sci Technol 19:055503
11. Tyo JS, Goldstein DL, Chenault DB, Shaw JA (2006) Review of passive imaging polarimetry for remote sensing applications. Appl Opt 45:5453–5469
12. Wolff LB, Boult TE (1991) Constraining object features using a polarisation reflectance model. IEEE Trans Pattern Anal Mach Intell 13:635–657
13. Wolff LB (1994) Diffuse-reflectance model for smooth dielectric surfaces. J Opt Soc Am A 11:2956–2968
14. Atkinson GA, Hancock ER (2007) Shape estimation using polarization and shading from two views. IEEE Trans Pattern Anal Mach Intell 29:2001–2017
15. Können GP (1985) Polarized light in nature. Cambridge University Press, Cambridge/New York
16. Schechner YY, Narashimhan SG, Nayar SK (2003) Polarization-based vision through haze. Appl Opt 42: 511–525
17. Schechner YY, Karpel N (2005) Recovery of underwater visibility and structure by polarization analysis. IEEE J Ocean Eng 30:570–587
18. Miyazaki D, Kagesawa M, Ikeuchi K (2004) Transparent surface modelling from a pair of polarization images. IEEE Trans Pattern Anal Mach Intell 26:73–82
19. Umeyama S, Godin G (2004) Separation of diffuse and specular components of surface reflection by use of polarization and statistical analysis of images. IEEE Trans Pattern Anal Mach Intell 26:639–647
20. Wallace AM, Liang B, Clark J, Trucco E (1999) Improving depth image acquisition using polarized light. Int J Comput Vis 32:87–109
21. Schechner YY, Shamir J, Kiryati N (2000) Polarization and statistical analysis of scenes containing a semireflector. J Opt Soc Am 17:276–284
22. Farid H, Adelson EH (1999) Separating reflections from images using independent components analysis. J Opt Soc Am 16:2136–2145
23. Atkinson GA, Hancock ER (2008) Two-dimensional BRDF estimation from polarisation. Comput Vis Image Underst 111:126–141
24. Lefaudeux N, Lechocinski N, Clemenceau P, Breugnot S (2009) New luster formula for the characterization of hair tresses using polarization imaging. J Cosmet Sci 60(2):153–169

# Polarizer

Daisuke Miyazaki
Graduate School of Information Sciences, Hiroshima City University, Asaminami-ku, Hiroshima, Japan

## Synonyms

Polarization filter; Polarizing film

## Related Concepts

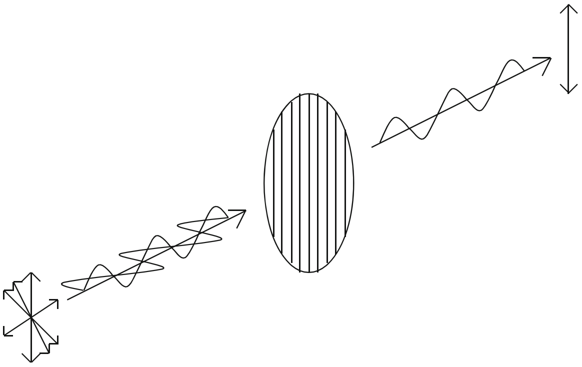
▶Polarization

## Definition

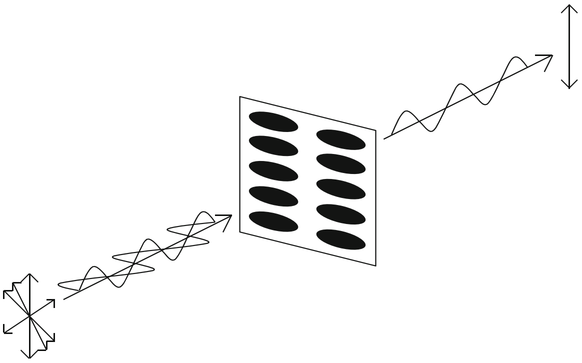
The device which polarizes the light when the light goes through it.

## Background

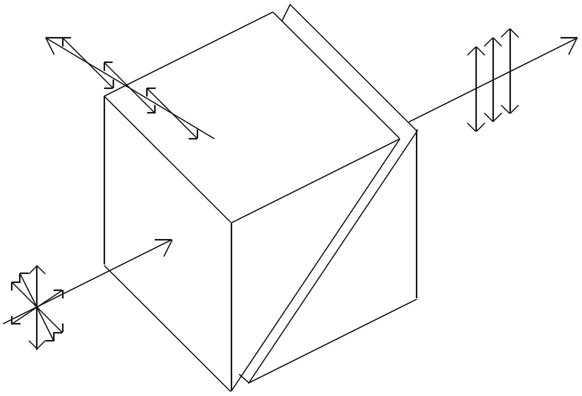
Polarizers makes the light oscillate in one direction. The polarization state of light changes if it hits any



**Polarizer, Fig. 1** Illustration of linear polarizer



**Polarizer, Fig. 2** Dichroic polarizer



**Polarizer, Fig. 3** Polarization prism which is called Glan Taylor Polarizer

materials, and polarization of the light is useful in many application fields. Polarizers are made artificially since natural objects usually do not change the light to be always perfectly polarized.

## Theory

There are two kinds of perfectly polarized light: linearly polarized light and circularly polarized light. This entry first explains linear polarizers, and then explains circular polarizers.

Figure 1 is the typical illustration of linear polarizer. Linear polarizer is often expressed as a circle with some straight lines inside. The orientation of the lines expresses the orientation of oscillation of the light transmitted through the polarizer. Figure 1 represents the case when the unpolarized light goes through the polarizer, and then, the electric vector field of the transmitted light oscillates only in vertical direction.

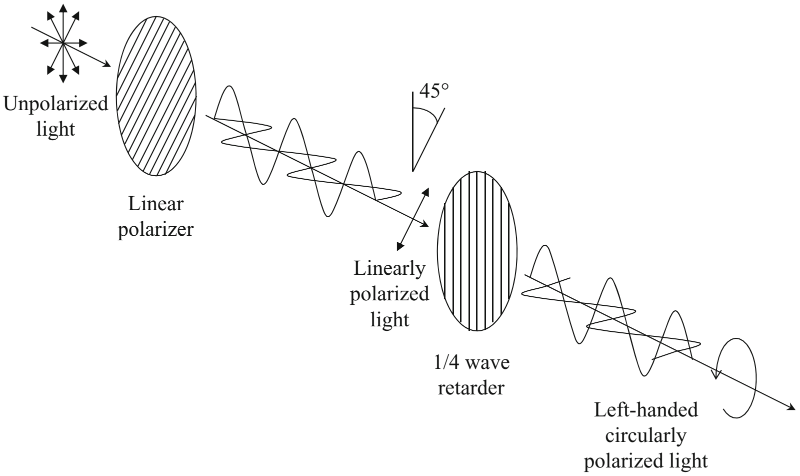
Typical polarizers are dichroic polarizers and wire-grid polarizers. The typical material used in dichroic polarizer is iodine. As is shown in Fig. 2, if the long axis of iodine molecule is lining up horizontally, the light transmitted through the polarizer will be vertically polarized. In this case, the horizontal component of the light is absorbed by the iodine molecule. The same applies to the wire-grid polarizers: If the wire-grid is alined horizontally, the transmitted light becomes vertically polarized.

Polarizers composed of prisms are also widely used. The typical material used in these polarizers is calcite. There are many types of prism polarizers such as Glan Taylor Polarizer, Glan Thompson Polarizer, Wollaston Polarizer, Rochon Polarizer, Nicol Polarizer, etc. Figure 3 is an illustration of Glan Taylor Polarizer. Two calcite prisms are separated by thin layer of the air. Calcite is a birefringent material, and the extraordinary ray is only transmitted through this polarizer.

Circularly polarized light is generated by circular polarizer. The circular polarizer is consisted of linear polarizer and 1/4 wave retarder as is shown in Fig. 4. The angle between the transmission axis of linear polarizer and the fast axis of 1/4 wave retarder is set to be 45°.

## Application

Liquid crystal display and liquid crystal projector use two linear polarizers in order to change the brightness of the light. In order to produce a three-dimensional view using projectors, polarized 3D glasses are often used. The viewer wears the polarized 3D glasses, and watch the screen. Two images are superimposed



**Polarizer, Fig. 4** Circular polarizer

into the screen; however, each eye perceives different image. This structure makes the viewer to recognize the 3D effect.

## References

1. Shurcliff WA (1962) Polarized light: production and use. Harvard University Press, Cambridge

# Polarizing Film

▸Polarizer

# Principal Axis

▸Optical Axis

# Principal Component Analysis (PCA)

Takio Kurita
Graduate School of Engineering, Hiroshima
University, Higashi-Hiroshima, Japan

## Synonyms

Karhunen–Loève transform (KLT)

## Definition

Principal component analysis (PCA) is a standard tool in modern data analysis and is used by almost all scientific disciplines. The goal of PCA is to identify the most meaningful basis to reexpress a given data set. It is expected that this new basis will reveal hidden structure in the data set and filter out the noise. There are so many applications such as dimensionality reduction, data compression, feature extraction, and data visualization.

## Background

The modern form of PCA was formalized by Hotelling [3] who also introduced the term *principal component*. It is also known as the Karhunen–Loève transform.

Observations are often described by several dependent variables which are, in general, intercorrelated and include noise. PCA is used to extract the important information from such observations and to reduce the noise. To achieve this goal, PCA computes a set of new orthogonal variables called *principal components* which are obtained as linear combinations of the original variables. The values of these new variables for the observations are called *factor scores*. They can be interpreted as the projections of the observations onto the principal components. PCA can be also formulated as the linear projection that minimizes the average projection cost defined as the mean squared distance between the data points and their projections [5].

Several reformulations or extensions of PCA have been proposed. PCA can be expressed as the maximum likelihood solution of a probabilistic latent variable model [6]. This reformulation is known as *probabilistic* PCA. A nonlinear generalization of PCA, which is known as kernel PCA, is also proposed by using the approach of kernel learning [8]. Sparse principal component analysis finds sparse coefficients by introducing a constraint on the norm of the coefficients [9]. This sparseness makes the interpretation of the results of PCA easier.

## Theory

Let $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]$ be the data set to be analyzed by PCA, where each column is a single observation described by $M$ variables. Then the sample mean vector $\bar{\boldsymbol{x}}$ and the sample covariance matrix $\Sigma$ can be represented by

$$\bar{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i \tag{1}$$

$$\Sigma = \frac{1}{N}\sum_{i=1}^{N} (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T = \frac{1}{N}\tilde{X}\tilde{X}^T \tag{2}$$

where the matrix $\tilde{X}$ is defined as $\tilde{X} = [\boldsymbol{x}_1 - \bar{\boldsymbol{x}}, \ldots, \boldsymbol{x}_N - \bar{\boldsymbol{x}}]$.

To extract the important information from the observations, PCA computes factor scores as linear combinations of the original variables

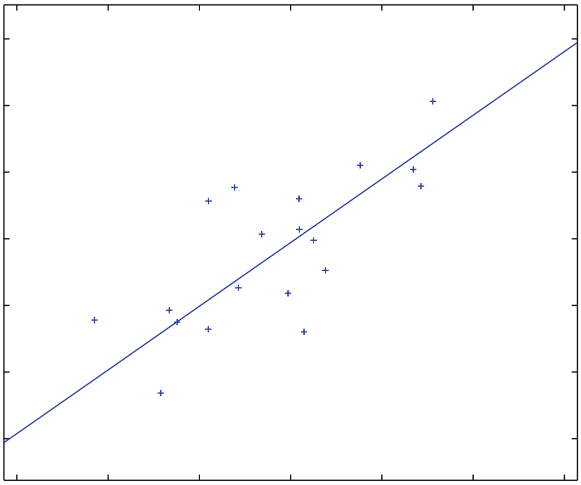
$$y_{1i} = \boldsymbol{a}_1^T(\boldsymbol{x}_i - \bar{\boldsymbol{x}}), \quad (i = 1, \ldots, N) \tag{3}$$

where $\boldsymbol{a}_1 = (a_{11}, \ldots, a_{M1})^T$ is a set of weights of the linear combinations. The optimum weight vector $\boldsymbol{a}_1$ is obtained so that the sample variance of the new variable is maximized under the normalization constraint $\boldsymbol{a}_1^T\boldsymbol{a}_1 = 1$. Since the sample variance is represented by

$$V(y_{11}, \ldots, y_{1N}) = \boldsymbol{a}_1^T\Sigma\boldsymbol{a}_1, \tag{4}$$

the optimization problem can be defined by using Lagrange multiplier $\lambda_1$ as the maximization of the Lagrange function

$$L(\boldsymbol{a}_1, \lambda_1) = \boldsymbol{a}_1^T\Sigma\boldsymbol{a}_1 - \lambda_1(\boldsymbol{a}_1^T\boldsymbol{a}_1 - 1). \tag{5}$$



**Principal Component Analysis (PCA), Fig. 1** An example of principal axis for two-dimensional samples

By setting the derivative of the Lagrange function with respect to the weight vector $\boldsymbol{a}_1$ equal to zero, the solution of this problem can be obtained as a unit eigenvector of the covariance matrix $\Sigma$ corresponding to the largest eigenvalue $\lambda_1$ as

$$\Sigma\boldsymbol{a}_1 = \lambda_1\boldsymbol{a}_1. \tag{6}$$

If the vector $\boldsymbol{a}_1$ is multiplied from the left, the maximum variance is given by

$$V(y_{11}, \ldots, y_{1N}) = \boldsymbol{a}_1^T\Sigma\boldsymbol{a}_1 = \lambda_1. \tag{7}$$

Figure 1 shows an example of the first principal axis for the two-dimensional samples.

The second principal components

$$y_{2i} = \boldsymbol{a}_2^T(\boldsymbol{x}_i - \bar{\boldsymbol{x}}), \quad (i = 1, \ldots, N) \tag{8}$$

can be defined as a new variable which maximizes the projected variance among all possible directions orthogonal to the first principal axis under the constraint on the normalization. The optimum coefficient vector $\boldsymbol{a}_2$ is also obtained as a unit eigenvector of the covariance matrix $\Sigma$ corresponding to the second largest eigenvalue $\lambda_2$. Similarly, additional principal components can be defined in an incremental fashion. For the general case of $L$-dimensional projection, the optimal linear projection of PCA

**Principal Component Analysis (PCA), Fig. 2** Examples of eigenfaces computed from 200 face images

$$Y = A^T \tilde{X} \tag{9}$$

can be obtained by taking the $L$ eigenvectors $A = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_L]$ of the covariance matrix $\Sigma$ corresponding to the $L$ largest eigenvalue $\lambda_1, \ldots, \lambda_L$. Then the eigenvector equation is given by

$$\Sigma A = A \Lambda, \tag{10}$$

where $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_L)$.

PCA is closely related with the singular value decomposition (SVD) of the data matrix. The data matrix $\tilde{X}$ can be decomposed by SVD as

$$\tilde{X} = S \Delta V^T, \tag{11}$$

where $S$ is the matrix of left singular vectors, $V$ is the matrix of right singular vectors, and $\Delta$ is the diagonal matrix of singular values. By using this relation, the covariance matrix can be rewritten as

$$\Sigma = \frac{1}{N} \tilde{X} \tilde{X}^T = \frac{1}{N} S \Delta V^T V \Delta S^T = \frac{1}{N} S \Delta^2 S^T. \tag{12}$$

By multiplying $S$ from the right, the eigenvector equation is obtained as

$$\Sigma S = S \frac{1}{N} \Delta^2. \tag{13}$$

This shows that the weight vectors $A$ is equal to $S$ and the diagonal matrix of the eigenvalues $\Lambda$ is equal to $\frac{1}{N}\Delta^2$ if the number of principal components $L$ is equal to the rank of the data matrix $\tilde{X}$. From these relations, the matrix of factor scores $Y$ can be represented as

$$Y = A^T \tilde{X} = S^T S \Delta V^T = \Delta V^T. \tag{14}$$

This shows that the principal components are also obtained from the SVD of the data matrix $\tilde{X}$. By using this relation, the data matrix $\tilde{X}$ can be represented as the product of the score matrix by the weight vectors $A$:

$$\tilde{X} = S \Delta V^T = A Y = A A^T \tilde{X}. \tag{15}$$

This means that the original data matrix $\tilde{X}$ can be reconstructed from the factor scores.

If the number of principal components $L$ is less than the rank of $\tilde{X}$, this reconstruction gives an approximation of the original data matrix. The mean squared errors between the original observations and the approximations are then given by

$$\begin{aligned} \varepsilon^2(L) &= \frac{1}{N} \sum_{i=1}^{N} ||(\boldsymbol{x}_i - \bar{x}) - A A^T (\boldsymbol{x}_i - \bar{x})||^2 \\ &= \sum_{i=1}^{\mathrm{rank}(\tilde{X})} \lambda_i - \sum_{i=1}^{L} \lambda_i \end{aligned} \tag{16}$$

which is simply the sum of the $rank(\tilde{X}) - L$ smallest eigenvalues. This means that PCA is minimizing this approximation errors by selecting the principal subspace spanned by the eigenvectors corresponding to the $L$ largest eigenvalues.

## Application

There are many applications of PCA in pattern recognition or computer vision. One of the famous applications of PCA is for face recognition [7]. Face images are decomposed into a set of characteristic feature images called "eigenfaces," which are the eigenvectors computed by PCA to the training set of face images. Recognition is performed by projecting a new face image into the subspace spanned by the eigenfaces and then classifying the face by comparing the distances of factor scores. Figure 2 shows examples of the eigenfaces computed from 200 face images.

PCA is also used to construct 3-D object models from their appearances [4]. To obtain a low-dimensional subspace of object appearance parametrized by pose and illumination, PCA is applied to a large set of images obtained by varying pose and illumination. Then a new image is projected to the constructed subspace and the recognition is performed on the subspace. The object's pose in the image is also determined from the position of the projection on the subspace.

## References

1. Adbi H, Williams LJ (2010) Principal component analysis. Wiley Interdiscip Rev Comput Stat 2:433–459
2. Bishop CM (2006) Pattern recognition and machine learning. Springer, New York
3. Hotelling H (1933) Analysis of a complex of statistical variables into principal components. J Educ Psychol 61:417–441
4. Murase H, Nayar S (1995) Visual learning and recognition of 3-D objects from appearance. Int J Comput Vis 14:5–24
5. Pearson K (1901) On lines and planes of closest fit to systems of points in space. Lond Edinb Dublin Philos Mag J Sci Sixth Ser 2:559–572
6. Tipping ME, Bishop CM (1999) Probabilistic principal component analysis. J R Stat Soc B 61:611–622
7. Turk M, Pentland A (1991) Eigenfaces for recognition. J Cogn Neurosci 3(1):71–86
8. Schölkopf B, Smola A, Müller KR (1998) Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput 10(5):1299–1319
9. Zou H, Hastie T, Tibshirani R (2006) Sparse principal component analysis. J Comput Graph Stat 15(2):265–286

# Principal Distance

►Focal length

# Probabilistic Hill Climbing

►Simulated Annealing

# Projection

Zhengyou Zhang
Microsoft Research, Redmond, WA, USA

## Related Concepts

►Affine Camera; ►Depth Distortion; ►Perspective Camera; ►Perspective Transformation; ►Weak Perspective Projection

## Definition

A projection is an image of a geometric figure, with or without its appearance property, reproduced on a line, a plane, or a surface. It is usually a mapping from a higher dimensional space to a lower dimensional subspace. A camera performs a projection from a 3D scene to a 2D picture. See related concepts listed above.

# Projection Matrix

►Projection Transformation

# Projection Transformation

Zhengyou Zhang
Microsoft Research, Redmond, WA, USA

## Synonyms

Projection matrix

## Related Concepts

►Affine Camera; ►Perspective Camera; ►Perspective Transformation

## Definition

*Projection transformation* describes the transformation from a 3D scene to a 2D image.

## Background

The actual projection transformation depends on the camera's projection model and is determined by the relationship between the center of projection (optical center) and the projection plane (image plane). If the center of projection is at a finite distance from the projection plane, it is a perspective transformation. If the center of projection is at infinity, the projection is parallel, known as an orthographic projection. There are several possible transformations between the two extremes, and the reader is referred to entry "▶Affine Camera" for more information.

In computer graphics a projection transformation transforms three-dimensional eye coordinates into points in three-dimensional *clip coordinates*. Besides various types of projection as mentioned above, it defines the *viewing volume*, which determines which objects or parts of objects are projected onto the screen. The viewing volume, also known as *viewing frustum*, is an six-sided enclosure defined by clipping planes: Primitives outside the viewing volume are not displayed.

# Projective Reconstruction

Richard Hartley
Department of Engineering, Australian National University, ACT, Australia

## Synonyms

Projective structure and motion

## Related Concepts

▶Euclidean Geometry; ▶Exploration: Simultaneous Localization and Mapping (SLAM)

## Definition

From several images of a scene and the coordinates of corresponding points identified in the different images, it is possible to construct a three-dimensional point-cloud model of the scene and compute the camera locations. From uncalibrated images the model can be reconstructed up to an unknown projective transformation, which can be upgraded to a Euclidean model by adding or computing calibration information.

## Background

Projective reconstruction refers to the computation of the structure of a scene from images taken with uncalibrated cameras, resulting in a scene structure, and camera motion that may differ from the true geometry by an unknown $3D$ projective transformation.

Suppose that a set of interest points are identified and matched (or tracked) in several images. The configuration of the corresponding $3D$ points and the locations of the cameras that took these images are supposed unknown. The task of reconstruction is to determine the values of these unknown quantities.

Formally, assume that a set of image points $\{\mathbf{x}_{ij}\}$ are known, where $\mathbf{x}_{ij}$ represents the image coordinates of the $j$-th point seen in the $i$-th image. It is generally not required that every point's location be known in every image, so only a subset of all possible $\mathbf{x}_{ij}$ are given. The structure-from-motion (SfM) problem is to determine the camera projection matrices $\mathtt{P}_i$ and the $3D$ point locations $\mathbf{X}_j$ such that the projection of the $j$-th point in the $i$th image is the measured $\mathbf{x}_{ij}$. Assuming a pinhole (projective) camera model, this relationship is expressed as a linear relationship:

$$\mathbf{x}_{ij} = \mathtt{P}_i \mathbf{X}_j \ , \qquad (1)$$

where $\mathtt{P}_i$ is a $3 \times 4$ matrix of rank 3, $\mathbf{X}_j$ and $\mathbf{x}_{ij}$ are expressed in homogeneous coordinates, and the equality is intended to hold only up to an unknown scale factor $\lambda_{ij}$. More precisely, therefore, the projection equation is

$$\lambda_{ij}\, \mathbf{x}_{ij} = \mathtt{P}_i \mathbf{X}_j \ . \qquad (2)$$

In the SfM problem, cameras $\mathtt{P}_i$ and points $\mathbf{X}_j$ are to be determined, given only the point correspondences.

### Homogeneous Coordinates

Both $2D$ (image) points and $3D$ (world) points are most conveniently expressed in *homogeneous coordinates*. Thus, an image point $\mathbf{x}$ is represented by a 3-vector $\mathbf{x} = (u, v, w)^\top$, known as its homogeneous representation. The relationship to the standard Euclidean (nonhomogeneous) coordinates $(x, y)$ of the point is given by $x = u/w$ and $y = v/w$. This process of division by the final coordinate of the homogeneous vector is known as *dehomogenization*. Note that two vectors $\mathbf{x} = (u, v, w)^\top$ and $\mathbf{x}' = (u', v', w')^\top$ represent the same point in Euclidean coordinates if and only if $\mathbf{x} = k\mathbf{x}'$ for some nonzero constant $k$. Thus, a given point may be expressed in infinitely many different ways in homogeneous coordinates. This is analogous with the way a given rational number has many different representations, such as $1/2 = 2/4 = 3/6 = k/2k$ for any $k$. One particularly convenient homogeneous representation of a point is the 3-vector with unit final coordinate: $(x, y, 1)^\top$.

Homogeneous coordinates (3-vectors) with final coefficient zero do not coincide to any real point in nonhomogeneous coordinates, since the process of dehomogenization involves division by zero. Such points are commonly known as points at infinity. The vector $(0, 0, 0)^\top$ is not considered to be a valid set of homogeneous coordinates.

In a similar way, $3D$ points are represented by homogeneous 4-vectors $\mathbf{X} = (\mathrm{X}, \mathrm{Y}, \mathrm{Z}, \mathrm{T})^\top$. The main advantage of using homogeneous coordinates to represent world and image points is that Eq. (1) has a particularly simple form as a linear relationship between the homogeneous coordinates of the points.

Two homogeneous vectors differing by a constant multiplicative factor are considered to be *equivalent* representations of the same point. The set of all equivalence classes of (nonzero) homogeneous $(n + 1)$-vectors form the *projective n-space,* $\mathcal{P}^n$. In studying projective reconstruction, it is conventional to consider image points to lie in projective 2-space $\mathcal{P}^2$, whereas $3D$ points lie in projective 3-space $\mathcal{P}^3$. This identifies the projective space $\mathcal{P}^2$ consisting of the (image) plane, augmented with points at infinity. Similarly, $\mathcal{P}^3$ consists of $\mathbb{R}^3$ along with a plane of points at infinity.

### Ambiguity

Expressed in full generality, the solution to the reconstruction problem may only be determined up to an unknown projective transformation, applied both to points and cameras.

A projective transformation of $\mathcal{P}^3$, the model for 3-space containing world points, is a mapping:

$$\mathbf{X} \mapsto \mathtt{H}\mathbf{X}$$

where $\mathtt{H}$ is a non-singular $4 \times 4$ matrix representing a mapping between homogeneous coordinates. Using this relationship, it is easily seen that the determination of camera matrices $\mathtt{P}_i$ and points $\mathbf{X}_j$ cannot be unique, given only corresponding image coordinates $\mathbf{x}_{ij}$. Consider

$$\begin{aligned} \mathbf{x}_{ij} &= \mathtt{P}_i \mathbf{X}_j \\ &= (\mathtt{P}_i \mathtt{H}^{-1})(\mathtt{H}\mathbf{X}_j) \\ &= \mathtt{P}'_i \mathbf{X}'_j \,. \end{aligned} \tag{3}$$

In this relationship, new points $\mathbf{X}'_j = \mathtt{H}\mathbf{X}_j$ are defined in terms of points $\mathbf{X}_j$ and similarly new camera matrices $\mathtt{P}'_i = \mathtt{P}_i \mathtt{H}^{-1}$ in terms of the camera matrices $\mathtt{P}_i$. Since both $(\{\mathtt{P}_i\}, \{\mathbf{X}_j\})$ and $(\{\mathtt{P}'_i\}, \{\mathbf{X}'_j\})$ give rise to the same projected image coordinates $\mathbf{x}_{ij}$, there is no way to choose between these two solutions to the reconstruction problem. In fact, there exists a complete family of solutions to the problem, corresponding to all possible choices of the matrix $\mathtt{H}$. All such solutions are related to each other by the application of a projective transformation and are hence called *projectively equivalent*. A particular solution, consisting of camera matrices $\mathtt{P}_i$ and points $\mathbf{X}_j$ satisfying Eq. (1) is known as a *projective reconstruction* of the scene, computed from the given corresponding image points.

The effect of projective ambiguity is given shown in Fig. 1.

### The Projective Reconstruction Theorem

The above analysis does not rule out the possibility that other solutions to this reconstruction problem exist, not related to a particular obtained solution by any projective transformation.

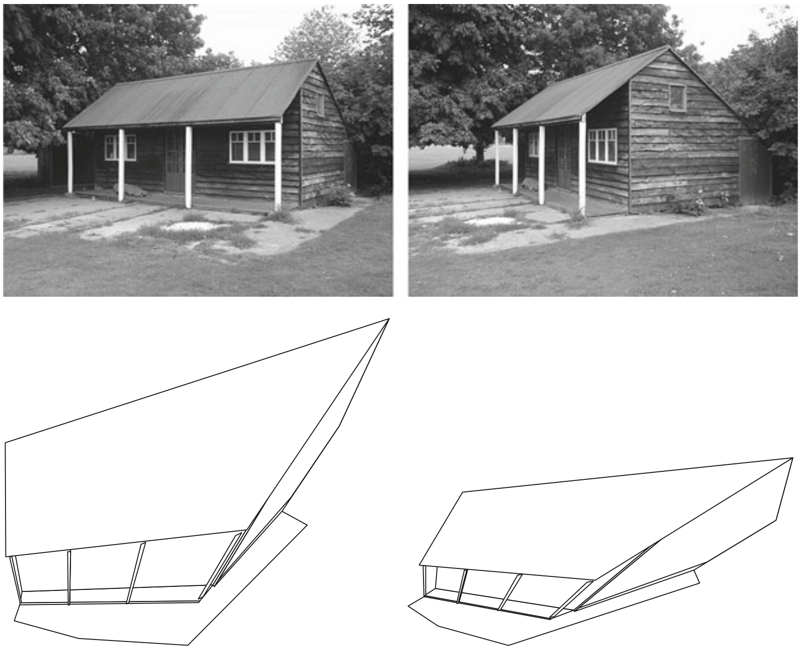
However, this possibility is excluded by the projective reconstruction theorem, which essentially says that if the set of corresponding points $\mathbf{x}_{ij}$ are sufficiently numerous (at least 8 in number), and do not lie in some degenerate configuration, then the solution to the reconstruction problem is unique up to a projective transformation.

The exact statement of the theorem requires the definition of the *fundamental matrix* which will be considered next.

## Theory and Applications

### Two View Reconstruction

Consider the reconstruction problem for only two images. Rather than using a subscript, entities belonging to the second camera are distinguished by a prime. Thus, the given input to this problem consists of corresponding points $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$; $i = 1, \ldots, n$, where the points $\mathbf{x}_i$ come from one image and the $\mathbf{x}'_i$ are the corresponding points in the other.



**Projective Reconstruction, Fig. 1** *Projective reconstruction.* (*Top*) Original image pair. (*Bottom*) Two views of a 3D projective reconstruction of the scene. The *lines* of the wireframe link the computed 3D points. The reconstruction requires no information about the camera matrices or information about the scene geometry. In a projective reconstruction, the resulting model is distorted by an arbitrary projective transformation from the true geometrically correct model (Figures derived from [15])

Let the camera matrices (unknown) be $\mathtt{P}$ and $\mathtt{P}'$, and let $\mathbf{X}_i$ be the $3D$ point corresponding to the image points $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$. The projection equations are

$$\begin{aligned} \lambda_i\,\mathbf{x}_i &= \mathtt{P}\mathbf{X}_i \\ \lambda'_i\,\mathbf{x}'_i &= \mathtt{P}'\mathbf{X}_i\,, \end{aligned}$$

where the scale factors $\lambda_i$ and $\lambda'_i$ are explicitly written (but are unknown). These equations may be written in a single system

$$\begin{bmatrix} \mathtt{P} & \mathbf{x}_i & \\ \mathtt{P}' & & \mathbf{x}'_i \end{bmatrix} \begin{pmatrix} \mathbf{X}_i \\ -\lambda_i \\ -\lambda'_i \end{pmatrix} = \mathbf{0}\,. \tag{4}$$

Since this equation must have a nonzero solution $(\mathbf{X}_i, -\lambda_i, -\lambda'_i)^\top$, the determinant of the matrix on the left (which shall be denoted as $\mathtt{A}$) must be zero. Since the point coordinates $\mathbf{x}_i$ and $\mathbf{x}'_i$ each appear in a single column, it follows that the determinant is a bilinear expression in $(\mathbf{x}_i, \mathbf{x}'_i)$, and hence, the equation $\det(\mathtt{A}) = 0$ can be written in the form

$$\mathbf{x}'^{\top}_i \mathtt{F}\mathbf{x}_i = 0\,, \tag{5}$$

where $\mathtt{F}$ is a $3 \times 3$ matrix depending only on the two camera matrices $\mathtt{P}$ and $\mathtt{P}'$. Consequently, this equation will hold for any pair of corresponding points $(\mathbf{x}_i, \mathbf{x}'_i)$. The matrix $\mathtt{F}$ is called the *fundamental matrix* corresponding to the camera pair $(\mathtt{P}, \mathtt{P}')$.

Closer examination of the matrix $\mathtt{A}$ appearing in (4) reveals the exact form of the matrix $\mathtt{F}$. Expanding $\det(\mathtt{A})$ by cofactors down the last two columns yields the following formula:

$$\mathtt{F}_{jk} = (-1)^{j+k} \det\begin{bmatrix} \mathtt{P}^{(\sim j)} \\ \mathtt{P}'^{(\sim k)} \end{bmatrix}\,, \tag{6}$$

where $\mathtt{P}^{(\sim j)}$ is the $2 \times 4$ matrix obtained by omitting the $j$th row of $\mathtt{P}$ and $\mathtt{P}'^{(\sim k)}$ is similarly defined.

Another way of writing the Eq. (5) is

$$(\mathbf{x}_i \otimes \mathbf{x}'_i)^\top \mathbf{f} = 0\,, \tag{7}$$

where $(\mathbf{x}_i \otimes \mathbf{x}'_i)^\top$ is the vector

$$(u'_i u_i,\ u'_i v_i,\ u'_i w_i,\ v'_i u_i,\ v'_i v_i,\ v'_i w_i,\ w'_i u_i,\ w'_i v_i,\ w'_i w_i) \tag{8}$$

expressed in terms of coordinates $\mathbf{x}_i = (u_i, v_i, w_i)^\top$ and $\mathbf{x}'_i = (u'_i, v'_i, w'_i)^\top$. Further, $\mathbf{f}$ is the vector $(\mathtt{F}_{11}, \mathtt{F}_{12}, \ldots, \mathtt{F}_{33})^\top$ made up of the entries of the fundamental matrix $\mathtt{F}$.

### Computing the Fundamental Matrix

Note that the Eq. (7) is a linear equation with unknowns equal to the entries of the fundamental matrix. The explicit form of the equation is given by (8). Given $n \geq 8$ point correspondences, one has a set of linear equations

$$\mathtt{A}\mathbf{f} = \mathbf{0},$$

where $\mathtt{A}$ is an $n \times 9$ matrix, with entries determined by the coordinates of the matched image points. This set of equations is solved to find $\mathbf{f}$.

Since this is a set of homogeneous equations, there is a solution $\mathbf{f} = \mathbf{0}$, which is not interesting; a nonzero solution is required. With exactly 8-point correspondences, there is an exact solution to this problem. With more points, a least-squares solution is computed. This is most conveniently done by solving the problem

$$\begin{aligned}\textbf{Minimize}\quad & \|\mathtt{A}\mathbf{f}\| \\ \textbf{subject to}\quad & \|\mathbf{f}\| = 1,\end{aligned}$$

where the condition $\|\mathbf{f}\| = 1$ is imposed in order to obtain a unique solution (apart from sign). The solution is the eigenvector of $\mathtt{A}^\top\mathtt{A}$ corresponding to the smallest eigenvalue. Alternatively, if $\mathtt{A}$ has singular value decomposition

$$\mathtt{A} = \mathtt{UDV}^\top,$$

then the required $\mathbf{f}$ is the last column of $\mathtt{V}$ (assuming that the singular values of $\mathtt{D}$ are in descending order). Once the solution $\mathbf{f}$ is found, the fundamental matrix $\mathtt{F}$ is reconstituted from the entries of $\mathbf{f}$.

The algorithm just described is the so-called 8-point algorithm for computing the fundamental matrix [20]. In order to get good results, it is necessary to preprocess the input image coordinates, using the so-called *normalized* 8-point algorithm, which will be described later.

### Projective Reconstruction Theorem

This discussion leads to the basic theorem of projective reconstruction, which states that under appropriate conditions, the reconstruction of a scene from sufficiently many point correspondences in two views is unique up to projective transformation.

**Theorem 1** *Let $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$; $i = 1, \ldots, n$ be point correspondences in two views and let $(\mathtt{P}, \mathtt{P}', \{\mathbf{X}_i\})$ be a pair of camera matrices, and some 3D points forming a 3D reconstruction specifically stated*

$$\begin{aligned}\lambda_i \mathbf{x}_i &= \mathtt{P}\mathbf{X}_i \\ \lambda'_i \mathbf{x}'_i &= \mathtt{P}'\mathbf{X}_i\end{aligned} \qquad (9)$$

*for some unknown $\lambda_i, \lambda'_i \neq 0$. Let $\mathtt{H}$ be an invertible $4 \times 4$ matrix $\mathtt{H}$, and define*

$$\begin{aligned}\tilde{\mathtt{P}} &= \mathtt{P}\mathtt{H}^{-1} \\ \tilde{\mathtt{P}}' &= \mathtt{P}'\mathtt{H}^{-1} \\ \tilde{\mathbf{X}}_i &= \mathtt{H}\mathbf{X}_i\ .\end{aligned} \qquad (10)$$

*Then the triple $(\tilde{\mathtt{P}}, \tilde{\mathtt{P}}', \{\tilde{\mathbf{X}}_i\})$ is also a reconstruction satisfying the Eq. (9).*

*Furthermore, if the set of vectors $\{\mathbf{x}_i \otimes \mathbf{x}'_i\}$ has rank 8 (spans a linear subspace of dimension 8 in $\mathcal{R}^9$), then any reconstruction $(\tilde{\mathtt{P}}, \tilde{\mathtt{P}}', \{\tilde{\mathbf{X}}_i\})$ satisfying (9) is related to the original reconstruction $(\mathtt{P}, \mathtt{P}', \{\mathbf{X}_i\})$ by (10) for some non-invertible matrix $\mathtt{H}$.*

This theorem was proved in [7, 16].

Note the condition that the set of vectors $\{\mathbf{x}_i \otimes \mathbf{x}'_i\}$ has rank 8 is exactly the condition that the set of equations of the form (7) has a unique solution. If the rank of the vectors $\{\mathbf{x}_i \otimes \mathbf{x}'_i\}$ is equal to 9, then there is no solution to the Eq. (7) and the point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ cannot be a valid set of points corresponding to projections of a set of $3D$ points in two images.

If the vectors $\{\mathbf{x}_i \otimes \mathbf{x}'_i\}$ span a space of dimension less than 8 (for instance, if there are fewer than 8-point correspondences), then there is not a unique matrix $\mathtt{F}$ satisfying the condition (5), and the reconstruction may not be unique up to projectivity.

### Extraction of Camera Matrices

Once the fundamental matrix has been computed, it is possible to extract a pair of camera matrices directly from $\mathtt{F}$. The decomposition is not unique, since according to Theorem 1, there are many pairs of camera matrices $(\mathtt{P}, \mathtt{P}')$ that correspond to the same fundamental matrix $\mathtt{F}$. It is always possible to assume that one of the camera matrices is of the form $\mathtt{P} = [\mathtt{I} \mid \mathbf{0}]$, so the problem is simply to compute the other camera matrix $\mathtt{P}'$.

An algorithm to do this is as follows:

1. Compute the singular value decomposition

$$\mathtt{F} = \mathtt{UDV}^\top,$$

where $\mathtt{D} \approx \mathrm{diag}(p, q, 0)$. Note that since $\mathtt{F}$ should have rank 2, the last singular value should be zero, but with noise this will not exactly hold.

2. Define matrices

$$\mathtt{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad ; \quad \mathtt{Z} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Define $\hat{\mathtt{D}} = \mathrm{diag}(p, q, r)$ for some value of $r$, and observe that

$$\mathtt{F} = \mathtt{UDU}^\top = (\mathtt{UZU}^\top)(\mathtt{UW}^\top\hat{\mathtt{D}}\mathtt{V}^\top) = \mathtt{S}\,\mathtt{M}\,, \quad (11)$$

where $\mathtt{S}$ is a skew-symmetric matrix and $\mathtt{M}$ is defined by this equation. The value of $r$ may be arbitrarily chosen.

3. A pair of camera matrices corresponding to the fundamental matrix $\mathtt{F}$ is now

$$\mathtt{P} = [\mathtt{I} \mid \mathbf{0}] \quad ; \quad \mathtt{P}' = [\mathtt{M} \mid \mathbf{u}_3] \qquad (12)$$

where $\mathbf{u}_3$ is the third column of $\mathtt{U}$.

### Notes

1. The vector $\mathbf{u}_3$ satisfies $\mathbf{u}_3^\top \mathtt{F} = \mathbf{u}_3^\top \mathtt{S} = \mathbf{0}$; it is the generator of the left null-space of $\mathtt{F}$.
2. The value of $r$, the last diagonal entry of $\hat{\mathtt{D}}$, may be chosen arbitrarily, but a good choice is to set $r = (p + q)/2$ so that $\mathtt{M}$ is far from singular.
3. If $r = 0$, the matrix $\mathtt{M}$ is singular, but has a particularly simple form, namely, $\mathtt{M} = \mathtt{SF}$. The corresponding camera $\mathtt{P}' = [\mathtt{SF} \mid \mathbf{u}_3]$ is sometimes used, but it has the property that the right-hand $3 \times 3$ block is singular, so the camera center lies at a nonfinite point.

### Complete Projective Reconstruction Algorithm

It is now possible to state a complete algorithm for projective reconstruction of a scene from two images. Suppose a set of image correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$; $i = 1, \ldots, n$ are given.

1. From the image correspondences, compute the fundamental matrix $\mathtt{F}$ linearly from Eq. (7), as described in section 3.1.
2. From $\mathtt{F}$ find the two camera projection matrices $\mathtt{P} = [\mathtt{I} \mid \mathbf{0}]$ and $\mathtt{P}' = [\mathtt{M} \mid \mathbf{t}]$, as in section 3.2.
3. The corresponding $3D$ points $\mathbf{X}_i$ may be computed linearly as the least-squares solution to Eq. (4). This process is called *triangulation*.

The linear triangulation method via Eq. (4) does not give optimal results. A method optimal in the presence of noise is given in [13, 14].

### The Normalized Eight-Point Algorithm

It was pointed out in [11] that the simple version of the 8-point algorithm given above can lead to very poor results in some circumstances, but this problem is largely alleviated by simple normalization of the image coordinates.

The issue with the 8-point algorithm for computing $\mathtt{F}$ is that the vector (Eq. 8) expressed in terms of image-point coordinates can contain entries of widely different magnitude. This leads to poor conditioning of the linear equations used to solve for $\mathtt{F}$. In addition, the results are dependent on the particular coordinate system (origin and scale) used to express image points.

Given corresponding image points $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$, one may define normalized coordinates $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{x}}'_i$ obtained from the original coordinates by the following operations:

1. Each $\mathbf{x}_i$ is replaced by $\mathbf{x}_i - \bar{\mathbf{x}}$, where $\bar{\mathbf{x}}$ is the mean (barycenter) of all the coordinates $\mathbf{x}_i$. This corresponds to a shift of the coordinate origin so that the mean of the $\mathbf{x}_i$ is at the origin.
2. The points are scaled so that their average (alternatively, their root-mean-squared) Euclidean distance from the origin is equal to $\sqrt{2}$. This is done by applying a common scaling to all the points $\mathbf{x}_i - \bar{\mathbf{x}}$. The resulting point is $\hat{\mathbf{x}}_i$.

The reason for choosing an average distance of $\sqrt{2}$ is so that the *average* point has homogeneous coordinates $(1, 1, 1)^\top$.

One applies these operations to the points $\mathbf{x}_i$ and $\mathbf{x}'_i$ independently. Note that both normalization steps are simple affine transformations of the points. These transformations may be written as

$$\hat{\mathbf{x}}_i = \mathtt{T}\mathbf{x}_i \quad ; \quad \hat{\mathbf{x}}'_i = \mathtt{T}'\mathbf{x}'_i \qquad (13)$$

where $\mathtt{T}$ and $\mathtt{T}'$ are $3 \times 3$ matrices acting on the homogeneous representations of the points.

Once this normalization has taken place, the computation of the fundamental matrix and the complete projective reconstruction may be carried out using the normalized coordinates. The result is a fundamental matrix $\hat{\mathtt{F}}$ satisfying the condition

$$\hat{\mathbf{x}}_i'^{\top}\hat{\mathtt{F}}\hat{\mathbf{x}}_i = 0 \tag{14}$$

from which by substitution using (13), one has

$$(\mathbf{x}_i'^{\top}\mathtt{T}'^{\top})\hat{\mathtt{F}}(\mathtt{T}\mathbf{x}_i) = 0 = \mathbf{x}_i'^{\top}\mathtt{F}\mathbf{x}_i \ .$$

From this it follows that $\mathtt{F} = \mathtt{T}'^{\top}\hat{\mathtt{F}}\mathtt{T}$ is the fundamental matrix corresponding to the original points.

Similarly, if $\hat{\mathtt{P}}$ and $\hat{\mathtt{P}}'$ are camera matrices belonging to a reconstruction from the normalized image coordinates, then

$$\hat{\mathbf{x}}_i = \hat{\mathtt{P}}\hat{\mathbf{X}}_i \ ; \ \hat{\mathbf{x}}_i' = \hat{\mathtt{P}}'\hat{\mathbf{X}}_i .$$

Once more, substituting for $\hat{\mathbf{x}}_i$ and $\hat{\mathbf{x}}_i'$, it follows that

$$\mathbf{x}_i = \mathtt{T}^{-1}\hat{\mathtt{P}}\hat{\mathbf{X}}_i \ ; \ \mathbf{x}_i' = \mathtt{T}'^{-1}\hat{\mathtt{P}}'\hat{\mathbf{X}}_i$$

which implies that the reconstruction $(\mathtt{P}, \mathtt{P}', \{\mathbf{X}_i\})$ for the original points $\mathbf{x}_i \leftrightarrow \mathbf{x}_i'$ is given by

$$\mathtt{P} = \mathtt{T}^{-1}\hat{\mathtt{P}} \ ; \ \mathtt{P}' = \mathtt{T}'^{-1}\hat{\mathtt{P}}' \ ; \ \mathbf{X}_i = \hat{\mathbf{X}}_i .$$

This normalized 8-point algorithm gives markedly superior results to the unnormalized algorithm, which should never be used directly. For more details and analysis, see [11].

## Three View Reconstruction

The 8-point algorithm and other methods involving the fundamental matrix are useful for reconstruction from two views.

If three images of a scene are available and point correspondences are known across all three views, then such linear methods can be extended to three-image reconstruction, using the *trifocal tensor*. This is an extension of the fundamental matrix to three views.

In this analysis of three-view reconstruction, it is convenient from a notational point of view to denote the three camera matrices as $\mathtt{A}$, $\mathtt{B}$, and $\mathtt{C}$, instead of $\mathtt{P}_1$, $\mathtt{P}_2$, and $\mathtt{P}_3$.

Given a three-way image-point correspondence $\mathbf{x}_i \leftrightarrow \mathbf{x}_i' \leftrightarrow \mathbf{x}_i''$, the goal is to find camera matrices $\mathtt{A}$, $\mathtt{B}$, and $\mathtt{C}$ and points $\mathbf{X}_i$ such that

$$\mathbf{x}_i = \mathtt{A}\mathbf{X}_i \ ; \ \mathbf{x}_i' = \mathtt{B}\mathbf{X}_i \ ; \ \mathbf{x}_i'' = \mathtt{C}\mathbf{X}_i . \tag{15}$$

This may be written in a form similar to (4), as follows:

$$\begin{bmatrix} \mathtt{A} & \mathbf{x}_i & & \\ \mathtt{B} & & \mathbf{x}_i' & \\ \mathtt{C} & & & \mathbf{x}_i'' \end{bmatrix} \begin{pmatrix} \mathbf{X}_i \\ -\lambda_i \\ -\lambda_i' \\ -\lambda_i'' \end{pmatrix} = \mathbf{0}. \tag{16}$$

In this case, the $9 \times 7$ matrix on the left is not square. Nevertheless, since there is a solution $(\mathbf{X}_i, -\lambda_i, -\lambda_i', -\lambda_i'')^{\top}$, the matrix must be rank-deficient. Consequently, any $7 \times 7$ submatrix must have vanishing determinant. Each such determinant implies a trilinear relationship between the coefficients of the matching points $\mathbf{x}_i \leftrightarrow \mathbf{x}_i' \leftrightarrow \mathbf{x}_i''$.

It is not necessary to consider all possible $7 \times 7$ submatrices to obtain useful relationships. Given three camera matrices $\mathtt{A}$, $\mathtt{B}$, and $\mathtt{C}$, one can define a triply indexed entity $\mathcal{T}_i^{qr}$:

$$\mathcal{T}_i^{qr} = (-1)^{i+1} \det \begin{bmatrix} \mathtt{A}^{(\sim i)} \\ \mathtt{B}^{(q)} \\ \mathtt{C}^{(r)} \end{bmatrix} . \tag{17}$$

Here, all indices range from 1 to 3. Further, $\mathtt{B}^{(q)}$ and $\mathtt{C}^{(r)}$ represent rows $q$ and $r$ of the matrices $\mathtt{A}$ and $\mathtt{B}$, whereas $\mathtt{A}^{(\sim i)}$ means the matrix $\mathtt{A}$ with row $i$ omitted. This results in a $4 \times 4$ matrix, whose determinant with the indicated sign is the chosen value $\mathcal{T}_i^{qr}$. This triply indexed set of 27 values is known as the *trifocal tensor* corresponding to the three cameras. Note that this tensor depends only on the camera matrices and not any image points.

Now, it may be shown [12, 15] that the coordinates of any matching triple $\mathbf{x}_i \leftrightarrow \mathbf{x}_i' \leftrightarrow \mathbf{x}_i''$ satisfy trilinear relations:

$$\sum_{i,j,k,q,r=1}^{3} x^i x'^j x''^k \epsilon_{jqu}\epsilon_{krv}\mathcal{T}_i^{qr} = 0_{uv}. \tag{18}$$

This relation is to be interpreted as follows:

1. The indices on the point coordinates, such as $x^i$, denote the $i$th coordinate of the homogeneous vector representing the point $\mathbf{x} = (x^1, x^2, x^3)^{\top}$.
2. The symbol $\epsilon_{jqu}$ (and similarly $\epsilon_{krv}$) has value 0 unless $j$, $q$, and $u$ are all distinct; otherwise, it is $+1$

if $jqu$ is an even permutation of the three indices 1, 2, and 3 and $-1$ if it is an odd permutation.

3. The indices $u$ and $v$ are free indices, and each choice of $u$ and $v$ leads to a different trilinear relation, for a total of 9 distinct relations. However, only 4 of these relations are linearly independent.

In the case where the first camera matrix $\mathtt{A}$ has the canonical form $[\mathtt{I} \mid \mathbf{0}]$, the expression (Eq. 17) for the trifocal tensor may be written simply as

$$\mathcal{T}_i^{qr} = b_i^q c_4^r - b_4^q c_i^r \,, \tag{19}$$

where $b_i^q$ is the element in row $q$ and column $i$ of $\mathtt{B}$ and $c_i^r$ is defined analogously.

A complete three-view reconstruction algorithm can then be outlined as follows:

1. From point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i \leftrightarrow \mathbf{x}''_i$ for $i = 1, \ldots, n$, each relation of the form (18) gives 4 linearly independent linear constraints on the entries of the trifocal tensor. From 7-point correspondences there are sufficiently many equations to compute $\mathcal{T}_i^{qr}$ linearly.
2. As with two-view reconstruction, it is possible to determine the form of the two other camera matrix $\mathtt{B}$ and $\mathtt{C}$ from the entries of the trifocal tensor using the formula (19).
3. Finally, by triangulation from three views based on the Eq. (16), one can find the image points $\mathbf{X}_i$, completing the reconstruction from three views.

A few more comments.

1. In the definition (18), the first camera matrix $\mathtt{A}$ is treated differently from the two others (in that two rows of $\mathtt{A}$ appear in the determinant, but only one from $\mathtt{B}$ and $\mathtt{C}$). There are two other similarly defined trifocal tensors in which matrices $\mathtt{B}$ or $\mathtt{C}$ are distinguished in this way.
2. Unlike with the fundamental matrix, there are relations similar to (18) that hold for line correspondences or mixed line and point correspondences. Thus, computation of the trifocal tensor and hence projective reconstruction is possible not only from point correspondences but from mixed correspondences of this type.

### Minimal Configurations

The reconstruction algorithms from two or three views described in Sections "Two View Reconstruction" and "Three View Reconstruction" require 8 or 7 points, respectively. However, it is possible to carry out reconstruction using only 7 points from 2 views, or as few as 6 points from 3 views.

From two views, the algorithm is easily explained. Given only 7-point correspondences, the set of equations $\mathbf{x}_i'^{\top} \mathtt{F} \mathbf{x}_i = 0$ represents a set of 7 homogeneous equations in the 9 entries of $\mathtt{F}$. The solution to this equation set is a two-parameter family $\mathtt{F} = \lambda \mathtt{F}_1 + \mu \mathtt{F}_2$ where $\mathtt{F}_1$ and $\mathtt{F}_2$ are determined by solving this system.

The condition that the fundamental matrix $\mathtt{F}$ must be a singular matrix gives a further equation $\det \mathtt{F} = 0$. Since $\mathtt{F}$ is a $3 \times 3$ matrix, this leads to a cubic homogeneous equation in $\lambda$ and $\mu$. Solving this cubic equation gives either one or three real solutions for the ratio $\lambda : \mu$ and hence one or three solutions (determined as ever up to scale) for the fundamental matrix $\mathtt{F}$. In short, from 7-point correspondences one or three possible fundamental matrices may be computed. From these possible values of $\mathtt{F}$, the rest of the method described previously will lead to a projective reconstruction, in fact either a unique or three possible reconstructions.

A method for computing the projective reconstruction from three views of 6 points is described in [27].

## Factorization Algorithms

The algorithms described previously for projective reconstruction work on two or three images. In many cases, one has many more images of a scene to use for reconstruction. To handle this situation, a variant of the Tomasi-Kanade factorization algorithm [30] may be used to do reconstruction from many views at once. This is the algorithm of Sturm and Triggs [29] for projective reconstruction.

As input, consider a set of image points $\mathbf{x}_{ij}$ for $i = 1, \ldots, m$ and $j = 1, \ldots n$, where $\mathbf{x}_{ij}$ represents the image of the $j$th point in the $i$th image. It is assumed (and required) that every point should be visible in every image, so $\mathbf{x}_{ij}$ is defined for all $(i, j)$.

The projection equations are of the form

$$\lambda_{ij}\, \mathbf{x}_{ij} = \mathtt{P}_i \mathbf{X}_j \,, \tag{20}$$

where the constants $\lambda_{ij}$ are required scale factors, the so-called *projective depths* of the points. This set of equations may be put together in one matrix equation as follows:

$$\begin{bmatrix} \lambda_{11}\mathbf{x}_{11} & \lambda_{12}\mathbf{x}_{12} & \dots & \lambda_{1n}\mathbf{x}_{1n} \\ \lambda_{21}\mathbf{x}_{21} & \lambda_{22}\mathbf{x}_{22} & \dots & \lambda_{2n}\mathbf{x}_{2n} \\ \vdots & & \ddots & \vdots \\ \lambda_{m1}\mathbf{x}_{m1} & \lambda_{m2}\mathbf{x}_{m2} & \dots & \lambda_{mn}\mathbf{x}_{mn} \end{bmatrix} = \begin{bmatrix} \mathtt{P}_1 \\ \mathtt{P}_2 \\ \vdots \\ \mathtt{P}_m \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 & \dots & \mathbf{X}_n \end{bmatrix} . \qquad (21)$$

In this equation the matrix on the left has dimension $3m \times n$, since each $\lambda_{ij}\mathbf{x}_{ij}$ is a 3-vector. This set of equations has the form

$$\Lambda \odot \mathtt{W} = \mathtt{PX} \qquad (22)$$

where

$$\Lambda = \begin{bmatrix} \lambda_{11} & \dots & \lambda_{1n} \\ \vdots & \ddots & \vdots \\ \lambda_{m1} & \dots & \lambda_{mn} \end{bmatrix} ; \mathtt{W} = \begin{bmatrix} \mathbf{x}_{11} & \dots & \mathbf{x}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{x}_{m1} & \dots & \mathbf{x}_{mn} \end{bmatrix} \qquad (23)$$

and $\odot$ is to be interpreted as an elementwise product, so that $\Lambda \odot \mathtt{W}$ is the matrix on the left of (21).

From the form of (21), it is evident that the matrix on the right has rank 4, since it is the product $\mathtt{PX}$ of matrices of dimension $3m \times 4$ and $4 \times n$. This is a low-rank constraint on the matrix $\Lambda \odot \mathtt{W}$ of depth-weighted image coordinates.

Unfortunately, although the matrix $\mathtt{W}$ of image coordinates is known, the matrix $\Lambda$ of projective depths is not known. With an incorrect set of projective depths, the matrix $\Lambda \odot \mathtt{W}$ will not have the expected rank 4. This suggests an algorithm in which the factorization and the projective depths are estimated alternately as follows:

1. Form the matrix $\mathtt{W}$ of homogeneous image coordinates as given in (Eq. 23), and define $\Lambda^0$ in which all $\lambda^0_{ij} = 1$. Then carry out the following steps iteratively for $k = 0, \dots, N$:
   (a) Find the closest rank-4 matrix $\mathcal{W}^k$ to $\Lambda^k \odot \mathtt{W}$.
   (b) Define $\Lambda^{k+1}$ to be the matrix of weights $\lambda^{k+1}_{ij}$ so that $\Lambda^{k+1} \odot \mathtt{W}$ is as close as possible to $\mathcal{W}^k$ under Frobenius norm.
2. Compute a final factorization $\mathcal{W}^N = \mathtt{PX}$, to obtain $\mathtt{P}$ and $\mathtt{X}$ providing the camera matrices and point locations, respectively.

In step 1(a), the low-rank factorization is carried out by singular value decomposition. Suppose $\Lambda^k \odot \mathtt{W} = \mathtt{UDV}^\top$. Let $\hat{\mathtt{D}}$ be the matrix obtained by setting all but the four first (largest) diagonal entries of $\mathtt{D}$ to zero. Then set $\mathcal{W}^k = \mathtt{U}\hat{\mathtt{D}}\mathtt{V}^\top$. The number of iterations $N$ is vaguely defined in this algorithm. The intention is to continue until "convergence," but as will be remarked below, continuing to convergence is problematic.

## Variants of the Method

It has been observed [24] that the bare projective algorithm as given above will converge to a *trivial* limit in which all the values of $\lambda_{ij}$ will be zero, except for those values in 4 columns of $\Lambda$. This solution is spurious, since zero values of the projective depths are not possible for a geometrically valid reconstruction. In addition, convergence is very slow. Therefore, different variants on the algorithm have been proposed, as follows:

1. In the original paper of Sturm and Triggs [29], an initialization of the projective depths is proposed, in which projective depths are derived from two-view reconstructions.
2. A viable strategy is to carry out only a fixed small number of alternation steps, since this significantly improves the solution without encountering a trivial solution.
3. A further step of normalization of the projective depths $\lambda_{ij}$ may be used [15]. Observe that if $\lambda_{ij}\mathbf{x}_{ij} = \mathtt{P}_i\mathbf{X}_j$, for all $(i, j)$, then for any constants $c_i$ and $d_j$,

   $$c_i d_j \lambda_{ij} \mathbf{x}_{ij} = (c_i \mathtt{P}_i)(d_j \mathbf{X}_j) . \qquad (24)$$

   Thus, each $\lambda_{ij}$ may be replaced by $c_i d_j \lambda_{ij}$ without materially changing the factorization. Thus, one may at will multiply each $i$th row of $\Lambda$ by $c_i$ and the $j$th column by a constant $d_j$. In [15] it is suggested that constants $c_i$ and $d_j$ may be chosen so that first the rows and then the columns of $\Lambda$ sum to unity. However, no analysis of this normalization procedure is given there.
4. More complex schemes for normalization schemes are given in [24] and [21, 22], for which convergence to a meaningful (local) minimum of some cost function is demonstrated.
5. Methods to accommodate missing data or outliers in projective factorization algorithms have been

proposed. Though many algorithms have addressed missing data in matrix factorization (for instance, [2–4, 18, 28]), a notable paper addressing projective factorization specifically is [5].

6. $L_1$-factorization has been recognized as more robust alternative to matrix factorization; an effective method is given in [6].

## Bundle Adjustment

Given measured image points $\mathbf{x}_{ij}$ in several images, the projection equations $\lambda_{ij}\mathbf{x}_{ij} = \mathtt{P}_i\mathbf{X}_j$ cannot be satisfied exactly if there is any inaccuracy, or noise, in the measurements. Therefore, in finding the projection matrices $\mathtt{P}_i$ and $3D$ points $\mathbf{X}_j$ to satisfy these equations, it is appropriate to find an approximate solution. Typically, this solution will be one that minimizes some appropriate cost function representing a residual error in the solution.

Since errors arise in the measurement of the coordinates of image points, it is appropriate to seek a solution that minimizes the error with respect to the measured image coordinates. This corresponds to choosing a cost function of the form

$$C(\{\mathbf{X}_j\}, \{\mathtt{P}_i\}) = \sum_{i,j\in\mathcal{N}} d(\mathbf{x}_{ij}, \mathtt{P}_i\mathbf{X}_j)^2 \, , \qquad (25)$$

where $\mathcal{N}$ is a set of pairs $(i, j)$ for which $\mathbf{x}_{ij}$ is measured. Further, $d(\mathbf{x}_{ij}, \mathtt{P}_i\mathbf{X}_j)$ represents the Euclidean distance in the two-dimensional image plane between the measured point $\mathbf{x}_{ij}$ and the projected point $\mathtt{P}_i\mathbf{X}_j$. This is commonly referred to as the *reprojection error*. The cost is to be minimized over all choices of $\mathtt{P}_i$ and $\mathbf{X}_j$. This is a nonlinear function. The choice of the squared distance means that a nonlinear least-squares cost function is to be minimized. The motivation for this choice is the observation that the solution to this least-squares problem represents the maximum likelihood (ML) solution, under the assumption that each image measurement error conforms to an isotropic Gaussian distribution, each point measurement being independent of the others.

Minimizing the cost function (25) over all choices of the variables $\mathtt{P}_i$ and $\mathbf{X}_j$ is known as *bundle adjustment*. Since this is a nonlinear optimization problem, an iterative algorithm is required. The most common algorithm used to minimize this cost function is the Levenberg-Marquardt algorithm [15, 19, 23, 31]. In order to converge to the globally optimal solution, a good initial solution is necessary. Such an initial solution is found by applying any of the algorithms previously described in this chapter.

### Robust Cost Functions

The cost function (25) is suitable and represents the ML solution if the measured point coordinates conform to a Gaussian distribution, and may be used if there are no gross errors (outliers) among the measured points. In most cases, this is unlikely, and a more robust cost function is to be preferred. In this case, the squared Euclidean distance function $d(\cdot,\cdot)^2$ is replaced by some other function $f(\cdot,\cdot)$ that is more tolerant of outliers, meaning that $f(\mathbf{x},\mathbf{y})$ grows less rapidly than $d(\mathbf{x},\mathbf{y})^2$ as the distance between the two arguments $\mathbf{x}$ and $\mathbf{y}$ increases. A good choice of robust cost function is the Huber cost function [15, 17]:

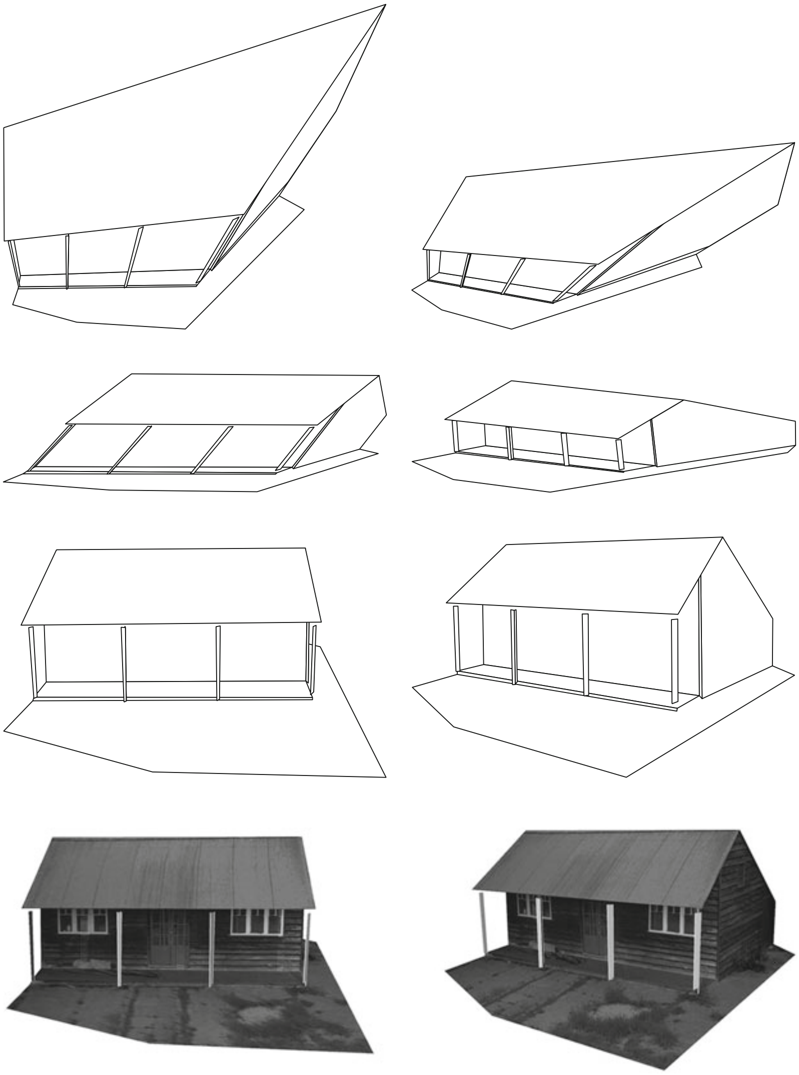
$$C(\{\mathbf{X}_j\}, \{\mathtt{P}_i\}) = \sum_{i,j\in\mathcal{N}} H\left(d(\mathbf{x}_{ij}, \mathtt{P}_j\mathbf{X}_i)\right)^2, \qquad (26)$$

where $H(x)^2$ is quadratic for $|x| < \delta$ and linear for $|x| \geq \delta$, and $\delta$ is some threshold approximately equal to the standard deviation of the measurements.

### Sparse Methods

A reasonable sized reconstruction problem may involve $1,000$ camera matrices $\mathtt{P}_i$ and $100,000$ points $\mathbf{X}_j$. Consequently, the cost function (25) depends on a large number of variables ($311,000$ parameters if the cameras are parametrized by 11 parameters). Since the central step in the Levenberg-Marquardt optimization process involves the solution of equations to compute the update of the parameters, this would involve solving a very large set of equations in all the variables. For a dense set of equations in $300,000$ parameters, this would be almost impossible.

Fortunately, the set of equations involved in this update process is quite sparse, so the problem is tractable. To see this, note that if a single point $\mathbf{X}_j$ is moved, then only the image points $\mathbf{x}_{ij}$ involving this point are affected. Similarly, if some camera matrix $\mathtt{P}_i$ is altered, then only image points $\mathbf{x}_{ij}$ are changed. This means that each image measurement depends only



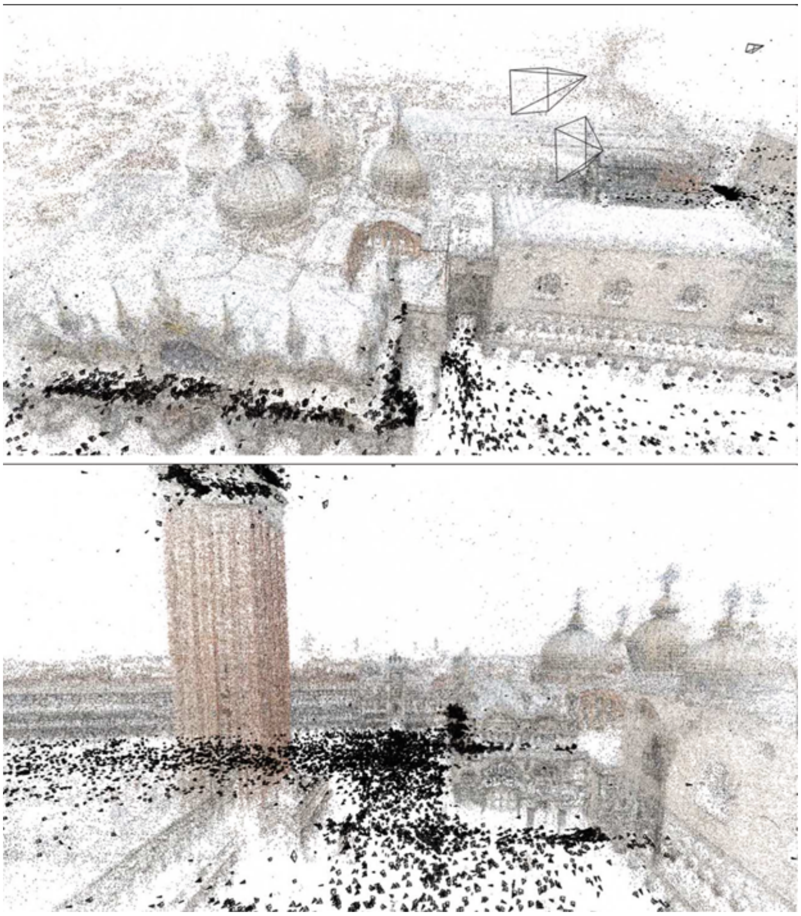
**Projective Reconstruction, Fig. 2** *Stratification*. The projective reconstruction (top row) obtained by uncalibrated reconstruction techniques is first upgraded to an affine reconstruction (second row). In the affine reconstruction, parallel lines in the image are parallel in the reconstruction, but geometric structures are still skewed. In the final stage of the reconstruction, the true Euclidean model (third row) is computed, in which angles and dimensions are correct up to an indeterminate scale. The fourth row shows two views of the texture-mapped model (Figures derived from [15])

on the parameters of one $3D$ point and one camera. This sparse dependence structure for the cost function results in a special sort of sparse structure for the Jacobian matrix. Sparse solution methods may then be used to accelerate the update step and allow it to be run in reasonable time. Methods that are used for this numerical problem include the Schurr complement method [15], in which the sparseness of the Jacobian is used to allow the camera updates to be computed first, followed by the point updates. The exact form of the equations is given in [15]. Alternatively, conjugate gradient methods [1] may be used; in such methods the sparseness of the equation set lends itself naturally to sparse methods.

## Euclidean Update

A projective reconstruction may be used as an initial step towards a geometrically correct (Euclidean) reconstruction. There are various ways in which this can be done:

1. By determining or knowing the calibration of the cameras. The camera calibration may be known a priori, or determined through the process of auto-calibration [9]. Constraints on the camera parameters, such as known focal length, or an assumption that some cameras have the same shared internal parameters, may be enforced easily during bundle adjustment. Automatic methods for auto-calibration



**Projective Reconstruction, Fig. 3** Views of reconstruction of San Marco, Venice, from Flickr images. The *top* image shows San Marco Cathedral and the doge's palace. Below is shown the campanile at *left* and the palace on the *right*. *Black* pyramids show the position and orientation of the cameras (Figures are reproduced with thanks to Noah Snavely)

often compute an affine reconstruction first, followed by an update to a Euclidean reconstruction and full determination of the camera calibration parameters [8, 10, 25]. This process is known as stratification.

2. By the knowledge of the $3D$ Euclidean coordinates of some number of *ground-control points*, at least five such points are required [16].
3. If partial camera calibration is known, the full calibration and Euclidean reconstruction may be computed more simply than if no calibration information is given. A notable paper demonstrating this is [26] and more details on self-calibration given different types of partial camera calibration are given in [15].

Figure 2 illustrates the steps from projective to Euclidean reconstruction via stratification.

A large-scale reconstruction, computed from thousands of images, is shown in Fig. 3.

## References

1. Agarwal S, Snavely N, Seitz SM, Szeliski R (2010) Bundle adjustment in the large. In: Proceedings of the 11th European conference on computer vision: Part II (ECCV'10). Springer, Berlin, pp 29–42
2. Brand M (2002) Incremental singular value decomposition of uncertain data with missing values. In: Proceedings of the 7th European conference on computer vision, part I (ECCV). Lecture notes in computer science, vol 2350, Copenhagen, Denmark. Springer, pp 707–720
3. Brandt S (2002) Closed-form solutions for affine reconstruction under missing data. In: Proceedings of the 7th European conference on computer vision, part I (ECCV). Lecture notes in computer science, vol 2350, Copenhagen, Denmark. Springer, pp 109–114
4. Buchanan AM, Fitzgibbon AW (2005) Damped Newton algorithms for matrix factorization with missing data. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), San Diego, vol 2, pp 316–322
5. Dai Y, Li H, He M (2010) Element-wise factorization for N-view projective reconstruction. In: Proceedings of

the European conference on computer vision (ECCV), Heraklion
6. Eriksson A, van den Hengel A (2010) Efficient computation of robust low-rank matrix approximations in the presence of missing data using the l1 norm. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), San Francisco, pp 771–778
7. Faugeras OD (1992) What can be seen in three dimensions with an uncalibrated stereo rig? In: Proceedings of the 2nd European conference on computer vision (ECCV), Santa Margharita Ligure, Italy, pp 563–578. Springer
8. Faugeras OD (1995) Stratification of three-dimensional vision: projective, affine, and metric representation. J Opt Soc Am A12:465–484
9. Faugeras OD, Luong Q, Maybank S (1992) Camera self-calibration: theory and experiments. In: Proceedings of the 2nd European conference on computer vision (ECCV), Santa Margharita Ligure, Italy. Springer, pp 321–334
10. Hartley RI (1994) Euclidean reconstruction from uncalibrated views. In: Mundy J, Zisserman A, Forsyth D (eds) Applications of invariance in computer vision. Lecture notes in computer science, vol 825. Springer, pp 237–256
11. Hartley RI (1997) In defense of the eight-point algorithm. IEEE Trans Pattern Anal Mach Intell 19(6):580–593
12. Hartley RI (1997) Lines and points in three views and the trifocal tensor. Int J Comput Vis 22(2):125–140
13. Hartley RI, Sturm P (1994) Triangulation. In: ARPA image understanding workshop, Monterey, pp 957–966
14. Hartley RI, Sturm P (1997) Triangulation. Comput Vis Image Underst 68(2):146–157
15. Hartley RI, Zisserman A (2004) Multiple view geometry in computer vision, 2nd edn. Cambridge University Press, Cambridge
16. Hartley RI, Gupta R, Chang T (1992) Stereo from uncalibrated cameras. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Champaign, pp 761–764
17. Huber PJ (1981) Robust statistics. Wiley, New York
18. Jacobs DW (2001) Linear fitting with missing data for structure-from-motion. Comput Vis Image Underst 82:57–81
19. Levenberg K (1944) A method for the solution of certain non-linear problems in least squares. Q Appl Math 2:164–168
20. Longuet-Higgins HC (1981) A computer algorithm for reconstructing a scene from two projections. Nature 293:133–135
21. Mahamud S, Hebert M (2000) Iterative projective reconstruction from multiple views. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Hilton Head, pp II–430–437
22. Mahamud S, Hebert M, Omori Y, Ponce J (2001) Provably-convergent iterative methods for projective structure from motion. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Kauai, pp 1018–1025
23. Marquardt DW (1963) An algorithm for least-squares estimation of nonlinear parameters. J Soc Ind Appl Math 11:431–441
24. Oliensis J, Hartley R (2007) Iterative extensions of the Sturm/Triggs algorithm: convergence and nonconvergence. IEEE Trans Pattern Anal Mach Intell 29(12): 2217–2233
25. Pollefeys M, Van Gool L (1999) Stratified self-calibration with the modulus constraint. IEEE Trans Pattern Anal Mach Intell 21(8):707–724
26. Pollefeys M, Koch R, Van Gool L (1998) Self calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In: Proceedings of the 6th international conference on computer vision, Bombay, India, pp 90–96
27. Quan L (1995) Invariants of six points and projective reconstruction from three uncalibrated images. IEEE Trans Pattern Anal Mach Intell 17:34–46
28. Shum HY, Ikeuchi K, Reddy R (1995) Principal component analysis with missing data and its application to polyhedral object modeling. IEEE Trans Pattern Anal Mach Intell 17(9):854–867
29. Sturm P, Triggs W (1996) A factorization based algorithm for multi-image projective structure and motion. In: Proceedings of the 4th European conference on computer vision (ECCV), Cambridge, pp 709–720
30. Tomasi C, Kanade T (1992) Shape and motion from image streams under orthography: a factorization approach. Int J Comput Vis 9(2):137–154
31. Triggs W, McLauchlan PF, Hartley RI, Fitzgibbon A (2000) Bundle adjustment for structure from motion. In: Triggs B, Zisserman A, Szeliski R (eds) Vision algorithms: theory and practice. Springer, Berlin, pp 298–372

## Projective Structure and Motion

▶Projective Reconstruction

## Prototype-Based Methods for Human Movement Modeling

Zhe Lin[1], Zhuolin Jiang[2] and Larry S. Davis[3]
[1]Advanced Technology Labs, Adobe Systems Incorporated, San Jose, CA, USA
[2]Noah's Ark Lab, Huawei Tech. Investment Co., LTD., Shatin, Hong Kong, China
[3]Computer Vision Laboratory, Center for Automation Research, University of Maryland, College Park, MD, USA

### Synonyms

Action prototype trees

## Definition

Human movement modeling is a static and dynamic human appearance representation with feature descriptors. Typical feature descriptors include local and holistic descriptors. Local descriptors mean sparse features extracted locally, i.e., local features [4] or local space-time features [6]; holistic descriptors means dense features extracted inside a human bounding region, i.e., shape/appearance descriptors [9], and motion descriptors [1, 3].

Prototype-based methods [2, 7, 13, 14] are a category of approaches representing original feature descriptors with a finite set of indices through feature quantization. Given a large number of descriptors extracted from training images or videos, a vector quantization (or data clustering) algorithm is used to divide the feature space into nonoverlapping cells where each cell is uniquely represented with an integer index. Given the quantization, each test feature can be mapped to an integer index (corresponding to the center of a quantization cell or cluster center) based on exact (or approximate) nearest neighbor search, and finally, recognition can be performed through simple index matching. This scheme is significantly more efficient than the direct descriptor matching schemes due to the speedup in calculating Euclidean distances in high-dimensional feature spaces.

## Background

There are two categories of methods to model human appearances and movements. The first category is local feature-based approaches. In this category, usually a set of local features are extracted across all image regions or video frames, and recognition is based on matching of those features either via histogram comparisons or class voting. For example, a bag-of-features representation is computed and used as the descriptors [11, 12], or a Hough voting-based approaches [15] are used to simultaneously locate humans and recognize their movements.

The second category is holistic feature-based approaches. These approaches typically extract a single high-dimensional feature descriptor from a hypothesized bounding region and classifies it with pre-trained classifier. Typically, a sliding window approach is used over all possible sub-rectangles across images [9] or across all space-time volumes [11, 14], and for every hypothetical bounding region, a holistic descriptor (consisting of appearance descriptor [9] and/or motion descriptor [1]) is extracted and classified into human/nonhuman (for human detection) or action IDs (for action recognition) based on a discriminative classifier trained off-line. The descriptors extracted from hypothetical human bounding region are invariant to human translation and scale variations and typically contain more information due to dense extraction; consequently, those features are much more reliable for recognition. In case of human movement understanding or activity recognition, typically, a generic human detector [9] is used to locate most probable human bounding boxes, and holistic descriptors extracted from all the frames are used for recognition.

Once feature descriptors are extracted from human images or videos, they will be fed into a classifier for recognition and categorization purposes. The classification modules are mostly based on machine learning or pattern recognition techniques as in the object recognition literature. Classifiers commonly used include NN/$k$-NN classifiers [2, 7], Support Vector Machine (SVM) classifiers [6], boosting-based classifiers [3, 5], dynamic time warping [13] for sequence alignment and matching, etc.

Direct descriptor matching and classification-based schemes have been common for human movement recognition. However, for large-scale action recognition problems, such a matching scheme may require tremendous amount of time for computing similarities between descriptor sequences and learning discriminative classifiers. Also, sequence matching (action recognition) involves frame alignment issues. In this regard, an efficient and effective descriptor matching scheme is needed to handle both scalability to the number of training classes and training examples and ability to handle sequence alignment.

## Theory

Prototype-based approaches have been very effective in handling scalability to large training data. These approaches typically represent a human action as a sequence of basic action units [2, 7, 13, 14]. Each action frame or frame segment (multiple consecutive frames) is represented as one of the holistic action units trained off-line or histogram of local action units (similar to bag of visual words). The action units are usually obtained by feature quantization such as $k$-means clustering in the feature space. The centers of quantization cells are defined as action units, and all test descriptors belong to a cell are assigned the same index (i.e., the index of the cell).

In [2], an action is represented as a set of pose primitives and $n$-Gram models are used for action matching. Weinland and Boyer [7] models an action as a set of minimum distances from exemplars to action frames in an exemplar-based embedding space. These action representation methods are compact and efficient but might be limited in capturing global temporal consistency between actions because they either use low-order statistics such as histograms and $n$-Grams or use a minimum-distance-based representation which does not enforce temporal ordering.

Relaxing temporal constraints [2] makes action representation more invariant to intra-class variation and, consequently, might be effective in recognizing small numbers of actions, but when the number of action classes is large, global temporal consistency is very important for action recognition due to small inter-class variability (i.e., increased ambiguity between actions). In fact, there have been approaches modeling the global temporal consistency.

### Prototype Tree-Based Method

Recently, tree-based methods have been popular for human detection and activity recognition problems. Mikolajczyk and Uemura [8] proposes a random forest method on large number local features for action recognition. Although this method is efficient due to fast nearest neighbor search and can handle action detection problems in difficult cases, action frame time alignment issue is not explicitly addressed. Lin et al. [13] learns an action prototype tree based on hierarchical k-means clustering [10] and aligns action sequences with a fast dynamic time warping algorithm in order to compute accurate similarities between action sequences. More specifically, the cluster centers of leaf nodes (of the tree) are defined as the set of action prototypes. Distances between action prototypes are precomputed and stored in a lookup table, so the sequence alignment and matching stage is very efficient. The action prototype tree-based method learns action prototypes in joint shape and motion space so that human movements can be described more accurately and compactly. This prototype tree model is also applied to sliding window-based framework for simultaneous action detection and recognition in [14]. The prototype tree-based motion modeling methods can also be combined with a Markov model to incorporate prototype transition priors between frames.

## Application

Appearance prototype-based methods generally can be applied to all the human-related recognition problems in computer vision including appearance-based human identification, human detection, and human action detection and recognition. More broadly, they can be applied to video surveillance, human-computer interaction, virtual reality, and multimedia understanding.

## Open Problems

Human movement modeling still remains challenging due to articulated nature of human bodies and varying camera viewpoints. Although there has been a significant progress on view-dependent appearance and movement modeling, view-invariant human movement modeling is still an open research topic. There has been several efforts to introduce view-invariant descriptors for human movements, but still more research is to be done to be practical in real applications.

## References

1. Efros AA, Berg AC, Mori G, Malik J (2003) Recognizing action at a distance. In: Proceedings of the international conference on computer vision (ICCV), Nice, pp 726–733
2. Thurau C, Hlavac V (2008) Pose primitive based human action recognition in videos or still images. In: Proceedings of the computer vision and pattern recognition (CVPR), Anchorage
3. Fathi A, Mori G (2008) Action recognition by learning mid-level motion features. In: Proceedings of the computer vision and pattern recognition (CVPR), Anchorage
4. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
5. Laptev I, Perez P (2007) Retrieving actions in movies. In: Proceedings of the international conference on computer vision (ICCV), Rio de Janeiro
6. Dollar P, Rabaud V, Cottrell G, Belongie S (2005) Behavior recognition via sparse spatio-temporal features. In: Proceedings of the VS-PETS, Beijing, pp 65–72

7. Weinland D, Boyer E (2008) Action Recognition Using Exemplar-based Embedding. In: Proceedings of computer vision and pattern recognition (CVPR), Anchorage
8. Mikolajczyk K, Uemura H (2008) Action Recognition with motion-appearance vocabulary forest. In: Proceedings of the computer vision and pattern recognition (CVPR), Anchorage
9. Dalal N, Triggs B (2005) Histograms of oriented gradients for Human detection. In: Proceedings of the computer vision and pattern recognition (CVPR), San Diego, pp 886–893
10. Nister D, Stewenius H (2006) Scalable recognition with a vocabulary tree. In: Proceedings of the computer vision and pattern recognition (CVPR), New York, NY, pp 2161–2168
11. Yuan J, Liu Z, Wu Y (2009) Discriminative Subvolume Search for Efficient Action Detection. In: Proceedings of the computer vision and pattern recognition (CVPR), Miami
12. Liu J, Luo J, Shah M (2009) Recognizing realistic actions from videos in the wild. In: Proceedings of the computer vision and pattern recognition (CVPR), Miami
13. Lin Z, Jiang Z, Davis LS (2009) Recognizing actions by shape-motion prototype trees. In: Proceedings of the international conference on computer vision (ICCV), Kyoto
14. Jiang Z, Lin Z, Davis LS (2010) A tree-based approach to integrated action localization, recognition and segmentation. In: Proceedings of the 3rd workshop on Human motion, Crete, Greece
15. Yao A, Gall J, Gool LV (2010) A hough transform-based voting framework for action recognition. In: Proceedings of the computer vision and pattern recognition (CVPR), San Francisco

## PSF Estimation

▸Blind Deconvolution

## PTZ Camera Calibration

▸Active Calibration

## Pulling a Matte

▸Matte Extraction

## Purposive Vision

▸Animat Vision